



## 3-Volt System Logic for Personal Computers

Data Book

Advanced  
Micro  
Devices



# 3-Volt System Logic for Personal Computers

## Data Book

ADVANCED MICRO DEVICES





© 1992 Advanced Micro Devices, Inc.

Advanced Micro Devices reserves the right to make changes in its products without notice in order to improve design or performance characteristics.

This publication neither states nor implies any warranty of any kind, including but not limited to implied warrants of merchantability or fitness for a particular application. AMD® assumes no responsibility for the use of any circuitry other than the circuitry in an AMD product.

The information in this publication is believed to be accurate in all respects at the time of publication, but is subject to change without notice. AMD assumes no responsibility for any errors or omissions, and disclaims responsibility for any consequences resulting from the use of the information included herein. Additionally, AMD assumes no responsibility for the functioning of undescribed features or parameters.

#### **Trademarks**

AMD is a registered trademark of Advanced Micro Devices, Inc.  
Am386, Flashrite, and GLITCH EATER are trademarks of Advanced Micro Devices, Inc.

Product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

---

## 3-VOLT SYSTEM LOGIC FOR PERSONAL COMPUTERS DATA BOOK

### Chapter 1

#### Am386™ FAMILY PRODUCTS

|  |      |
|--|------|
| Am386DXLV Microprocessor Data Sheet .....                                | 1-3  |
| Am386SXLV Microprocessor Data Sheet .....                                | 1-35 |
| Am386DXLV and Am386SXLV Microprocessors Technical Reference Manual ..... | 1-67 |

### Chapter 2

#### SMALL COMPUTER SYSTEM INTERFACE (SCSI) PRODUCTS

|                             |      |
|-----------------------------|------|
| Am53C94LV Data Sheet .....  | 2-3  |
| Am53CF94LV Data Sheet ..... | 2-17 |

### Chapter 3

#### ONE TIME PROGRAMMABLE (OTP) EPROM PRODUCTS

|                            |      |
|----------------------------|------|
| Am27LV512 Data Sheet ..... | 3-3  |
| Am27LV010 Data Sheet ..... | 3-17 |
| Am27LV020 Data Sheet ..... | 3-33 |



The desktop revolution has made the personal computer an indispensable business tool. Now, with the recent introduction of 3-volt ICs, designers are breaking the barrier to persuasive portable PC use by extending the useful battery life without weighing down their products. As manufacturers strike the ideal balance between power requirements, useful battery life and weight, users will have virtually unlimited portable machine choices.

AMD's announcement of the world's first 3-volt Am386™ microprocessor in October 1991, plus the five other products described in this data book, encouraged designers to optimize designs to produce the battery powered products *customers really want*. In addition, over 50 other companies have announced various 3-volt components.

AMD recognizes that 3-volt operation alone does not make the ultimate notebook. For this reason, the Am386SXLV and Am386DXLV microprocessors support software power management schemes such as Microsoft's Advanced Power Management (APM) specification and the System Management Mode (SMM) of operation. APM allows the system BIOS, operating system, and APM-compliant applications to share critical power management data while preserving compatibility between hardware and software. SMM allows system hardware or software to interrupt the CPU in order to efficiently control devices and peripherals further reducing overall power consumption. The combination of APM and SMM plus the overall power savings from 3-volt operation is opening new markets.

The Am386SXLV and Am386DXLV microprocessors provide the enabling technology for a new generation of portable systems, demonstrating AMD's commitment to accelerate the rate of IC developments for new markets.



# **Chapter 1**

## **Am386 Family Products**

## CHAPTER 1

### Am386 Family Products

|  |      |
|--|------|
| Am386DXLV Microprocessor Data Sheet .....                                | 1-3  |
| Am386SXLV Microprocessor Data Sheet .....                                | 1-35 |
| Am386DXLV and Am386SXLV Microprocessors Technical Reference Manual ..... | 1-67 |

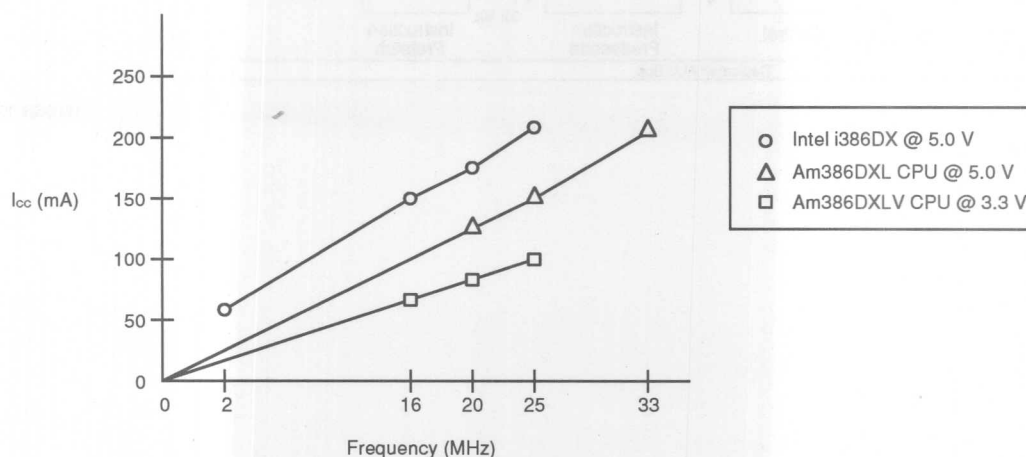


# Am386DXLV

## High-Performance, Low-Voltage, 32-Bit Microprocessor

### DISTINCTIVE CHARACTERISTICS

- **Operating voltage range 3.0 V to 5.5 V—Ideal for portable PC applications**
  - 25-MHz operating frequency for 3.0 V to 5.5 V
  - 33-MHz operating frequency for 4.5 V to 5.5 V
  - Twice the battery life of existing 5-V designs
  - Wide range of chipsets and other logic available for 3-V systems with support for Standby Mode operation
  - True static design for long battery life
  - Power consumption 85% lower than Intel i386DX, 65% lower than Am386DXL processor
  - Performance on demand (0 to 33 MHz)
- **SMM (System Management Mode) for system and power management**
  - SMI (System Management Interrupt) for power management independent of processor operating mode and operating system
  - SMI coupled with I/O instruction break feature provides transparent power off and auto resume of peripherals which may not be "power aware"
- SMI is non-maskable and has higher priority than Non-Maskable Interrupt (NMI)
- Automatic save and restore of the microprocessor state
- Wide range of chipsets supporting SMM available to allow product differentiation
- **Lower heat dissipation for fanless systems**
- **"Float" input to facilitate system debug and test**
- **Compatible with 386DX systems and software**
- **Supports 387DX-compatible math coprocessors**
- **132-pin PQFP package with optional protective ring for better lead coplanarity**
- **AMD® advanced 0.8 micron CMOS technology**



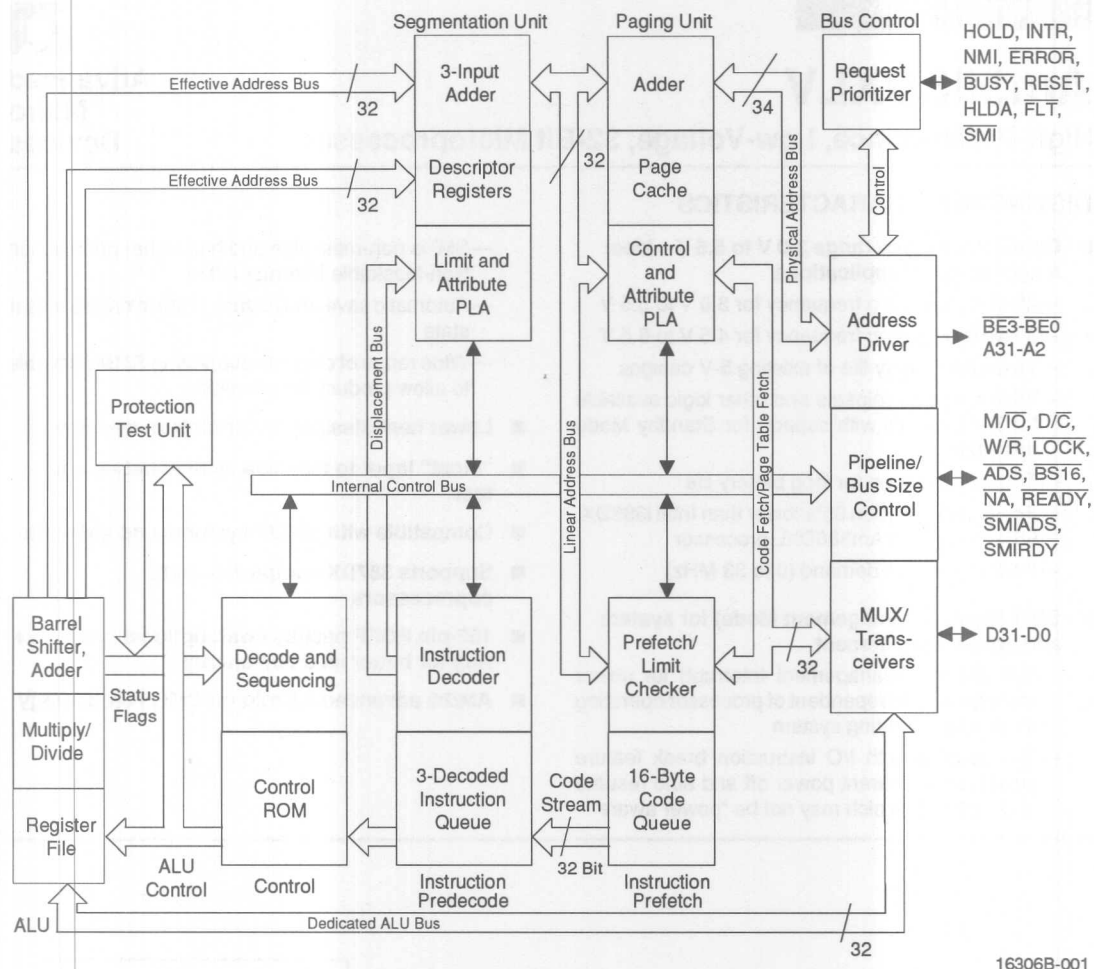
Note: Inputs at  $V_{CC}$  or  $V_{SS}$ .

### Typical Power Consumption

This document contains information on a product under development at Advanced Micro Devices, Inc. The information is intended to help you evaluate this product. AMD reserves the right to change or discontinue work on this proposed product without notice.



# BLOCK DIAGRAM



16306B-001

## GENERAL DESCRIPTION

The Am386DXLV microprocessor is a low-voltage, true static implementation of the Intel i386DX. The operating voltage range is 3.0 V to 5.5 V. The low-voltage operation makes the Am386DXLV microprocessor ideal for both desktop and battery-powered portable personal computers. For desktop PCs, low heat dissipation allows the system designers to remove or reduce the size and cost of the system cooling fan. The Am386DXLV microprocessor operates at a maximum speed of 25 MHz from 3.0 to 5.5 V and at a maximum speed of 33 MHz from 4.5 to 5.5 V.

The Am386DXLV microprocessor's lower operating voltage and true static design enables longer battery life and/or lower weight for portable applications. At

25 MHz, this device has 80% lower operating  $I_{cc}$  than the Intel i386DX. Lowering typical operating voltage from 5.0 V to 3.3 V doubles the battery life. Standby Mode allows the Am386DXLV microprocessor to be clocked down to 0 MHz (DC) and retain full register contents. In Standby Mode, typical current draw is 0.01 mA, a greater than 1000X reduction in power consumption versus the Intel i386DX.

The Am386DXLV processor is available in a small footprint 132-pin Plastic Quad Flat Pack (PQFP) package. This surface-mount package is 40% smaller than a PGA package, allowing smaller lower-cost board designs without the need for a socket.

Additionally, the Am386DXLV processor comes with SMM for system and power management. SMI is a non-maskable, higher priority interrupt than NMI and has its own code space (1 Mb in Real Mode and 4 Gb in Protected Mode). SMI can be coupled with the I/O instruction break feature to implement transparent power management of peripherals. SMM can be used by system designers to implement system and power management code independent of the operating system or the processor mode.

The Am386DXLV processor incorporates a float pin that places all outputs in a three-state mode to facilitate board test and debug.

## FUNCTIONAL DESCRIPTION

### Benefits of Lower Operating Voltage

The Am386DXLV microprocessor has an operating voltage range of 3.0 V to 5.5 V. Low voltage allows for lower operating power consumption, longer battery life, and/or smaller batteries for notebook applications.

Because power is proportional to the square of the voltage, reduction of the supply voltage from 5.0 V to 3.3 V reduces power consumption by 56%. This directly translates to a doubling of battery life for portable applications. Lower power consumption can also be used to reduce the size and weight of the battery. Thus, 3.3-V designs facilitate a reduction in the form factor.

A lower operating voltage results in a reduction of I/O voltage swings. This reduces noise generation and provides a less hostile environment for board design. Lower operating voltage also reduces electromagnetic radiation noise, and makes FCC approval easier to obtain.

### SMM—System Management Mode

The Am386DXLV processor has a new System Management Mode (SMM) for system and power management. This mode consists of two features: System Management Interrupt (SMI) and I/O instruction break.

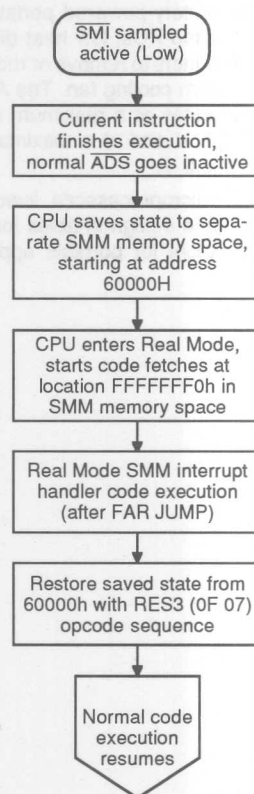
#### SMI—System Management Interrupt

SMI is implemented by using special bus interface pins. This interrupt method can be used to perform system management functions such as power management independent of processor operating mode (Real, Protected, or Virtual 8086 Modes).

SMI can also be invoked in software. This allows system software to communicate with SMI power management code. In addition, a UMOV instruction allows data transfers between SMI and normal system memory spaces.

Activating the  $\overline{\text{SMI}}$  pin invokes a sequence that saves the operating state of the processor into a separate SMM memory space, independent of the main system memory. After the state is saved, the processor is forced into Real Mode and begins execution at address FFFFFFF0h where a far jump to the SMM code is executed. This Real Mode code can perform its system

management function and then resume execution of the normal system software by executing an RES3 instruction that will reload the saved processor state and continue execution in the main system memory space. See Figure 1 for a general flowchart of an SMM operation.



16306B-002

Figure 1. SMM Flow

#### CPU Interface—Pin Functions

The CPU interface for SMM consists of three pins dedicated to the SMI function. One pin,  $\overline{\text{SMI}}$ , is the new interrupt input. The other two pins,  $\overline{\text{SMIADS}}$  and  $\overline{\text{SMIRDY}}$ , provide the control signals necessary for the separate SMI Mode memory space.

#### Description of SMM Operation

The execution of a System Management Interrupt has four distinct phases: the initiation of the interrupt via  $\overline{\text{SMI}}$ , a processor state save, execution of the SMM interrupt code, and a processor state restore (to resume normal operation).

#### Interrupt Initiation

A System Management Interrupt is initiated by the driving of a synchronous, active Low pulse on the  $\overline{\text{SMI}}$  pin until the first  $\overline{\text{SMIADS}}$  is asserted. This pulse period will ensure recognition of the interrupt. The CPU drives the



$\overline{\text{SMI}}$  pin active Low after the completion of the current operation (active bus cycle, instruction execution, or both). The active drive of the pin by the CPU is released at the end of the interrupt routine, following the last register read of the saved state. The CPU will drive  $\overline{\text{SMI}}$  High for two CLK2 cycles prior to releasing the drive of  $\overline{\text{SMI}}$ .

An SMI cannot be masked off by the CPU, and it will always be recognized by the CPU, regardless of operating mode. This includes the Real, Protected, and Virtual 8086 Modes of the processor.

While the CPU is in SMI Mode, a bus hold request via the HOLD pin is granted. The HLDA pin goes active after bus release and the  $\overline{\text{SMIADS}}$  pin floats along with the other pins that normally float during a bus hold cycle. The  $\overline{\text{SMI}}$  pin does not float during Bus Hold cycles.

### Processor State Save

The first set of SMM bus transfer cycles after the CPU's recognition of an active SMI is the processor saving its state to an external RAM array in a separate address space from main system memory. This is accomplished by using the  $\overline{\text{SMIADS}}$  and  $\overline{\text{SMIRDY}}$  pins for initiation and termination of bus cycles, instead of the  $\overline{\text{ADS}}$  and  $\overline{\text{READY}}$  pins. The 32-bit addresses to which the CPU saves its state are 60000H–600CBh and 60100h–60127h. These are fixed address locations for each register saved.

Pipelining must be disabled by asserting the  $\overline{\text{NA}}$  pin High in SMM. For  $\overline{\text{ADS}}$  initiated bus cycles,  $\overline{\text{BS16}}$  can be used. The value of BS16 is ignored for  $\overline{\text{SMIADS}}$  initiated bus cycles.

The save state takes approximately 630 CLK2 cycles with zero wait state memory. There are 61 data transfer cycles.

### SMI Code Execution

After the processor state is saved to the separate SMM memory space, the execution of the SMI interrupt routine code begins. The processor enters Real Mode, sets most of the register values to "reset" values (those values normally seen after a CPU reset), and begins fetching code from address FFFFFFF0h in the separate SMM memory space. Normally, the first thing the interrupt routine code does is a FAR JUMP to the Real Mode entry point for the SMI interrupt routine, which is also in SMM memory space.

INTR and NMI are disabled upon entry to SMM. See the Am386DXLV/Am386SXLV Microprocessor Technical Manual for further details on interrupts in SMM. The SMM code can be located anywhere within the 1-Mb Real Mode address space, except where the processor state is saved. I/O cycles, as a result of the IN, OUT, INS, and OUTS instructions, will go to the normal address space, utilizing the normal  $\overline{\text{ADS}}$  and  $\overline{\text{READY}}$  bus interface signals. This facilitates power management code manipulating system hardware registers as

needed through the standard I/O subsystem; a separate I/O space is not implemented.

### Processor State Restore (Resuming Normal Execution)

Returning to normal code execution in the main system memory, including restoring the processor operating mode, is accomplished by executing a RES3 (0Fh 07h) instruction. This instruction invokes a restore CPU state operation that reloads the CPU registers from the saved data in the RAM controlled by  $\overline{\text{SMIADS}}$  and  $\overline{\text{SMIRDY}}$ .

The ES:EDI register pair must point to the physical address of the saved state, this is normally at 60000h. In Real Mode, the address is calculated as  $\text{ES} \cdot 16 + \text{EDI}$ . The saved state should not cross a 64K boundary. The special opcode sequence RES3 should be executed to start the restore state operation. After completion of the restore state operation, the  $\overline{\text{SMI}}$  pin will be deactivated by the CPU and normal code execution will continue at the point where it left off before the SMI occurred.

In a zero wait state memory implementation, approximately 574 CLK2 cycles complete the restore state operation. There are 61 data transfer cycles.

### Software Features

There are several features of the SMI function that provide support for special operations during the execution of the system's software. These features involve the execution of reserved opcodes to induce specific SMI related operations.

### Software SMI Generation

Besides hardware initiation of the System Management Interrupt via the  $\overline{\text{SMI}}$  pin, there is also a software induced SMI mechanism. Generating a soft SMI involves setting a control bit 12 in Debug Register 7 (DR7) and executing an SMI instruction (reserved opcode F1h).

The functional sequence of the software-based SMI is identical to the hardware-based SMI with the exception that the  $\overline{\text{SMI}}$  pin is not initially driven active by an external source. Upon execution of a soft SMI opcode, the  $\overline{\text{SMI}}$  pin is driven active (Low) by the processor before the save state operation begins.

### Memory Transfers to Main System Memory

While executing an SMI routine, the interrupt code can initiate memory data reads and writes to the main system memory using the normal  $\overline{\text{ADS}}$  and  $\overline{\text{READY}}$  pins. This is accomplished by using reserved opcodes that are special forms of the MOV instruction (called UMOV). The UMOV opcodes can move byte, word, or doubleword register operands to or from main system memory. Multiple data transfers using the normal  $\overline{\text{ADS}}$  and  $\overline{\text{READY}}$  pins will occur if the operands are misaligned relative to the effective address used. The UMOV opcodes are 0Fh 10h, 0Fh 11h, 0Fh 12h, and 0Fh 13h.

The UMOV instruction can use any of the 386 addressing modes, as specified in the ModR/M byte of the op-

code. Note that the 16- and 32-bit versions are the same opcodes with the exception of the 66h operand size prefix. The  $\overline{BS16}$  line is recognized during the normal memory space data transfer(s) initiated by these instructions.

### I/O Instruction Break

The Am386DXLV processor has an I/O instruction break feature that allows the system logic to implement I/O trapping for peripheral devices. To enable the I/O instruction break feature,  $\overline{IIBEN}$  must first be asserted active Low. On detecting an I/O instruction, the processor prevents the execution unit from executing further instructions until  $\overline{READY}$  is driven active Low by the system. Once  $\overline{READY}$  is driven active, the execution unit either immediately responds to any active interrupt request or continues executing instructions following the I/O instruction that caused the break.

The I/O instruction break feature can be used to allow system logic to implement I/O trapping for peripheral devices. On sensing an I/O instruction, the system drives the  $\overline{SMI}$  interrupt active before driving  $\overline{READY}$  active. This ensures that the service routine is executed immediately following the I/O instruction that caused the break. (If the I/O instruction break feature is not enabled via  $\overline{IIBEN}$ , several instructions that follow the I/O instruction that caused the break will execute before the SMI service routine is executed.) The SMI service routine can access the peripheral for which SMI was asserted and modify the peripheral's state.

The SMI service routine normally returns to the instruction following the I/O instruction that caused the break. By modifying the saved state instruction pointer, the routine can choose to return to the I/O instruction that caused the break and re-execute that instruction. The default is to return to the following instruction (except for REP I/O strings). To re-execute the I/O instruction that caused the break, the SMI service routine must copy the I/O instruction pointer over the default pointer. This feature is particularly useful when an application program requests an access to a peripheral that has been powered down. The SMI service routine can restore power to the peripheral and initiate a re-execution sequence transparent to the application program. This re-execution feature should only be used if the SMI is in response to an I/O trap with  $\overline{IIBEN}$  active.

Note that the I/O instruction break feature is not enabled for memory mapped I/O devices or for 80387 bus cycles, even if  $\overline{IIBEN}$  is active.

### I/O Instruction Break Timing

The I/O Instruction Break feature requires that  $\overline{SMI}$  be sampled active (Low) by the processor at least three CLK2 edges before the CLK2 edge that ends the I/O cycle with an active  $\overline{READY}$  signal. This timing applies

for both pipelined and non-pipelined cycles. If this timing constraint is not met, additional instructions may be executed by the internal execution unit prior to entering SMI Mode.

Depending on the state of the prefetch queue at the time  $\overline{SMI}$  is asserted, instruction fetch cycles may occur on the normal ADS interface before the SMI save state process begins with the assertion of  $\overline{SMIADS}$ . However, this fetched code will not be executed.

### True Static Operation

The Am386DXLV microprocessor incorporates a true static design. Unlike dynamic circuit design, the Am386DXLV device eliminates the minimum operating frequency restriction. It may be clocked from its maximum speed all the way down to 0 MHz (DC). System designers can use this feature to design true 32-bit battery-powered portable PCs with long battery life.

### Standby Mode

This true static design of the Am386DXLV microprocessor allows for a Standby Mode. At any of its operating speeds, the Am386DXLV microprocessor will retain its state (i.e., the contents of all of its registers). By shutting off the clock completely, the device enters Standby Mode. Since power consumption is a function of clock frequency, operating power consumption is reduced as the frequency is lowered. In Standby Mode, typical current draw is reduced to less than 0.01 mA at DC. This feature not only saves battery life, but it also simplifies the design of power-conscious notebook computers in the following ways.

1. Eliminates the need for software in BIOS to save and restore the contents of registers.
2. Allows simpler circuitry to control stopping of the clock since the system does not need to know the state of the processor.

### Lower Operating Icc

True static design also allows lower operating Icc when operating at any speed.

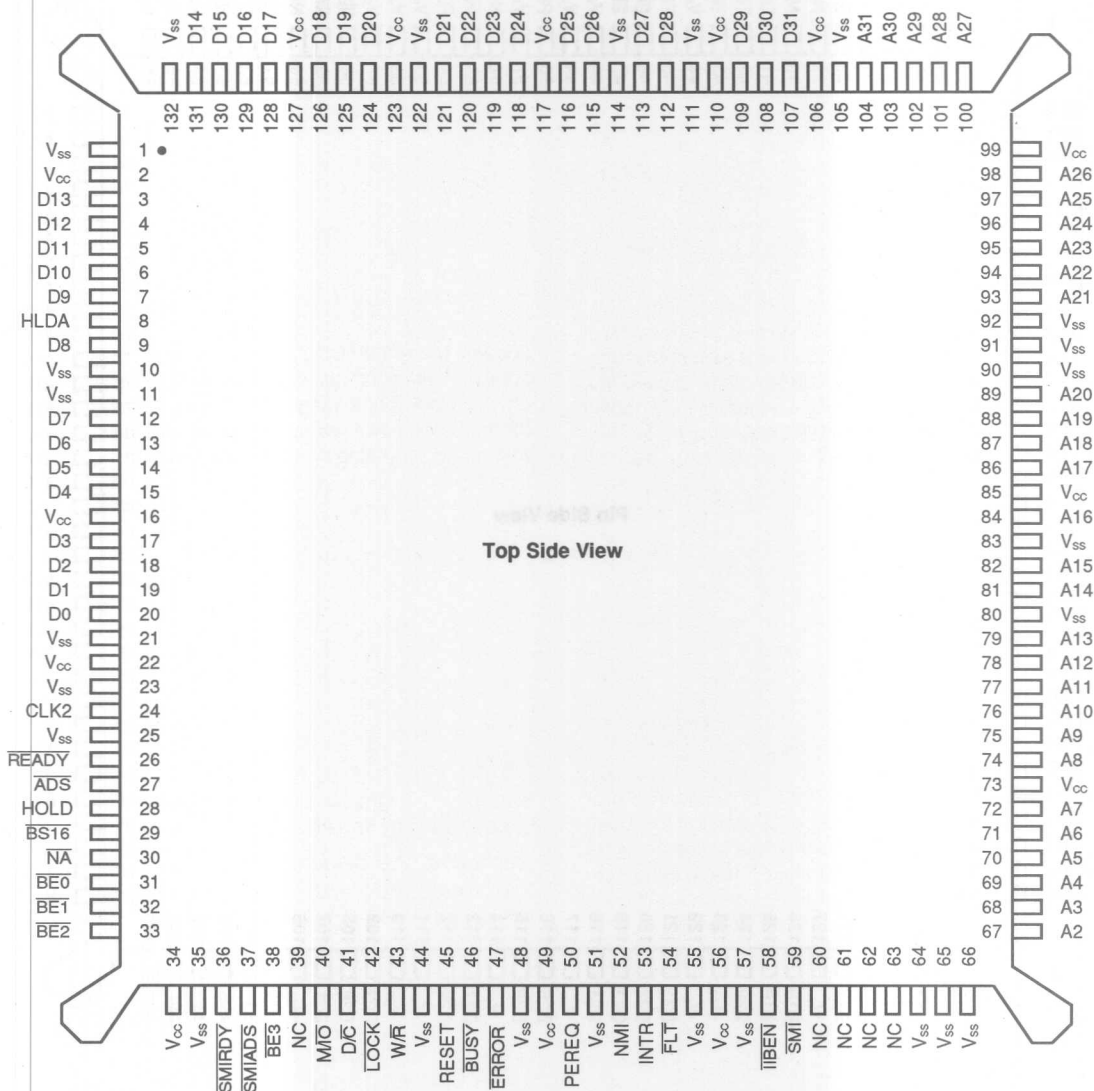
### Performance on Demand

The Am386DXLV microprocessor retains its state at any speed from 0 MHz (DC) to its maximum operating speed. With this feature, system designers may vary the operating speed of the system to extend the battery life in portable systems.

For example, the system could operate at low speeds during inactivity or polling operations. However, upon interrupt, the system clock can be increased up to its maximum speed. After a user-defined time-out period, the system can be returned to a low (or 0 MHz) operating speed without losing its state. This design maximizes battery life while achieving optimal performance.

## CONNECTION DIAGRAM

## 132-Lead Plastic Quad Flat Pack (PQFP) Package — Top Side View



Notes: Pin 1 is marked for orientation.

NC = Not connected; connection of an NC pin may cause a malfunction or incompatibility with future shippings of the Am386DXLV microprocessor.



# CONNECTION DIAGRAM

## 132-Lead Plastic Quad Flat Pack (PQFP) Package — Pin Side View



Notes: Pin 1 is marked for orientation.

NC = Not connected; connection of an NC pin may cause a malfunction or incompatibility with future shippings of the Am386DXLV microprocessor.

## PQFP Pin Designations (Sorted by Functional Grouping)

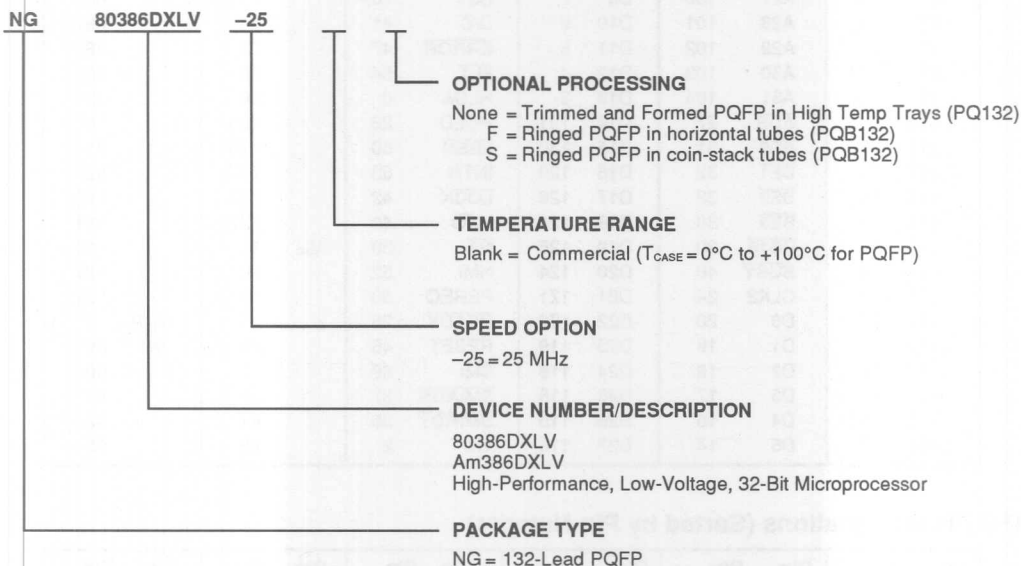
| Pin Name | Pin No. | Pin Name | Pin No. | Pin Name | Pin No. | Pin Name        | Pin No. | Pin Name        | Pin No. | Pin Name        | Pin No. |
|----------|---------|----------|---------|----------|---------|-----------------|---------|-----------------|---------|-----------------|---------|
| A2       | 67      | A24      | 96      | D6       | 13      | D28             | 112     | V <sub>CC</sub> | 16      | V <sub>SS</sub> | 51      |
| A3       | 68      | A25      | 97      | D7       | 12      | D29             | 109     |                 | 22      |                 | 55      |
| A4       | 69      | A26      | 98      | D8       | 9       | D30             | 108     |                 | 34      |                 | 57      |
| A5       | 70      | A27      | 100     | D9       | 7       | D31             | 107     |                 | 49      |                 | 64      |
| A6       | 71      | A28      | 101     | D10      | 6       | D/C             | 41      |                 | 56      |                 | 65      |
| A7       | 72      | A29      | 102     | D11      | 5       | ERROR           | 47      |                 | 73      |                 | 66      |
| A8       | 74      | A30      | 103     | D12      | 4       | FLT             | 54      |                 | 85      |                 | 80      |
| A9       | 75      | A31      | 104     | D13      | 3       | HLDA            | 8       |                 | 99      |                 | 83      |
| A10      | 76      | ADS      | 27      | D14      | 131     | HOLD            | 28      |                 | 106     |                 | 90      |
| A11      | 77      | BE0      | 31      | D15      | 130     | IIBEN           | 58      |                 | 110     |                 | 91      |
| A12      | 78      | BE1      | 32      | D16      | 129     | INTR            | 53      |                 | 117     |                 | 92      |
| A13      | 79      | BE2      | 33      | D17      | 128     | LOCK            | 42      |                 | 123     |                 | 105     |
| A14      | 81      | BE3      | 38      | D18      | 126     | M/IO            | 40      |                 | 127     |                 | 111     |
| A15      | 82      | BS16     | 29      | D19      | 125     | NA              | 30      | V <sub>SS</sub> | 1       |                 | 114     |
| A16      | 84      | BUSY     | 46      | D20      | 124     | NMI             | 52      |                 | 10      |                 | 122     |
| A17      | 86      | CLK2     | 24      | D21      | 121     | PEREQ           | 50      |                 | 11      |                 | 132     |
| A18      | 87      | D0       | 20      | D22      | 120     | READY           | 26      |                 | 21      | W/R             | 43      |
| A19      | 88      | D1       | 19      | D23      | 119     | RESET           | 45      |                 | 23      | NC              | 39      |
| A20      | 89      | D2       | 18      | D24      | 118     | SMI             | 59      |                 | 25      |                 | 60      |
| A21      | 93      | D3       | 17      | D25      | 116     | SMIADS          | 37      |                 | 35      |                 | 61      |
| A22      | 94      | D4       | 15      | D26      | 115     | SMIRDY          | 36      |                 | 44      |                 | 62      |
| A23      | 95      | D5       | 14      | D27      | 113     | V <sub>CC</sub> | 2       |                 | 48      |                 | 63      |

## PQFP Pin Designations (Sorted by Pin Number)

| Pin No. | Pin Name        | Pin No. | Pin Name        | Pin No. | Pin Name        | Pin No. | Pin Name        | Pin No. | Pin Name        | Pin No. | Pin Name        |
|---------|-----------------|---------|-----------------|---------|-----------------|---------|-----------------|---------|-----------------|---------|-----------------|
| 1       | V <sub>SS</sub> | 23      | V <sub>SS</sub> | 45      | RESET           | 67      | A2              | 89      | A20             | 111     | V <sub>SS</sub> |
| 2       | V <sub>CC</sub> | 24      | CLK2            | 46      | BUSY            | 68      | A3              | 90      | V <sub>SS</sub> | 112     | D28             |
| 3       | D13             | 25      | V <sub>SS</sub> | 47      | ERROR           | 69      | A4              | 91      | V <sub>SS</sub> | 113     | D27             |
| 4       | D12             | 26      | READY           | 48      | V <sub>SS</sub> | 70      | A5              | 92      | V <sub>SS</sub> | 114     | VSS             |
| 5       | D11             | 27      | ADS             | 49      | V <sub>CC</sub> | 71      | A6              | 93      | A21             | 115     | D26             |
| 6       | D10             | 28      | HOLD            | 50      | PEREQ           | 72      | A7              | 94      | A22             | 116     | D25             |
| 7       | D9              | 29      | BS16            | 51      | V <sub>SS</sub> | 73      | V <sub>CC</sub> | 95      | A23             | 117     | V <sub>CC</sub> |
| 8       | HLDA            | 30      | NA              | 52      | NMI             | 74      | A8              | 96      | A24             | 118     | D24             |
| 9       | D8              | 31      | BE0             | 53      | INTR            | 75      | A9              | 97      | A25             | 119     | D23             |
| 10      | V <sub>SS</sub> | 32      | BE1             | 54      | FLT             | 76      | A10             | 98      | A26             | 120     | D22             |
| 11      | V <sub>SS</sub> | 33      | BE2             | 55      | V <sub>SS</sub> | 77      | A11             | 99      | V <sub>CC</sub> | 121     | D21             |
| 12      | D7              | 34      | V <sub>CC</sub> | 56      | V <sub>CC</sub> | 78      | A12             | 100     | A27             | 122     | V <sub>SS</sub> |
| 13      | D6              | 35      | V <sub>SS</sub> | 57      | V <sub>SS</sub> | 79      | A13             | 101     | A28             | 123     | V <sub>CC</sub> |
| 14      | D5              | 36      | SMIRDY          | 58      | IIBEN           | 80      | V <sub>SS</sub> | 102     | A29             | 124     | D20             |
| 15      | D4              | 37      | SMIADS          | 59      | SMI             | 81      | A14             | 103     | A30             | 125     | D19             |
| 16      | V <sub>CC</sub> | 38      | BE3             | 60      | NC              | 82      | A15             | 104     | A31             | 126     | D18             |
| 17      | D3              | 39      | NC              | 61      | NC              | 83      | V <sub>SS</sub> | 105     | V <sub>SS</sub> | 127     | V <sub>CC</sub> |
| 18      | D2              | 40      | M/IO            | 62      | NC              | 84      | A16             | 106     | V <sub>CC</sub> | 128     | D17             |
| 19      | D1              | 41      | D/C             | 63      | NC              | 85      | V <sub>CC</sub> | 107     | D31             | 129     | D16             |
| 20      | D0              | 42      | LOCK            | 64      | V <sub>SS</sub> | 86      | A17             | 108     | D30             | 130     | D15             |
| 21      | V <sub>SS</sub> | 43      | W/R             | 65      | V <sub>SS</sub> | 87      | A18             | 109     | D29             | 131     | D14             |
| 22      | V <sub>CC</sub> | 44      | V <sub>SS</sub> | 66      | V <sub>SS</sub> | 88      | A19             | 110     | V <sub>CC</sub> | 132     | V <sub>SS</sub> |

**ORDERING INFORMATION**
**Standard Products**

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of the elements below.

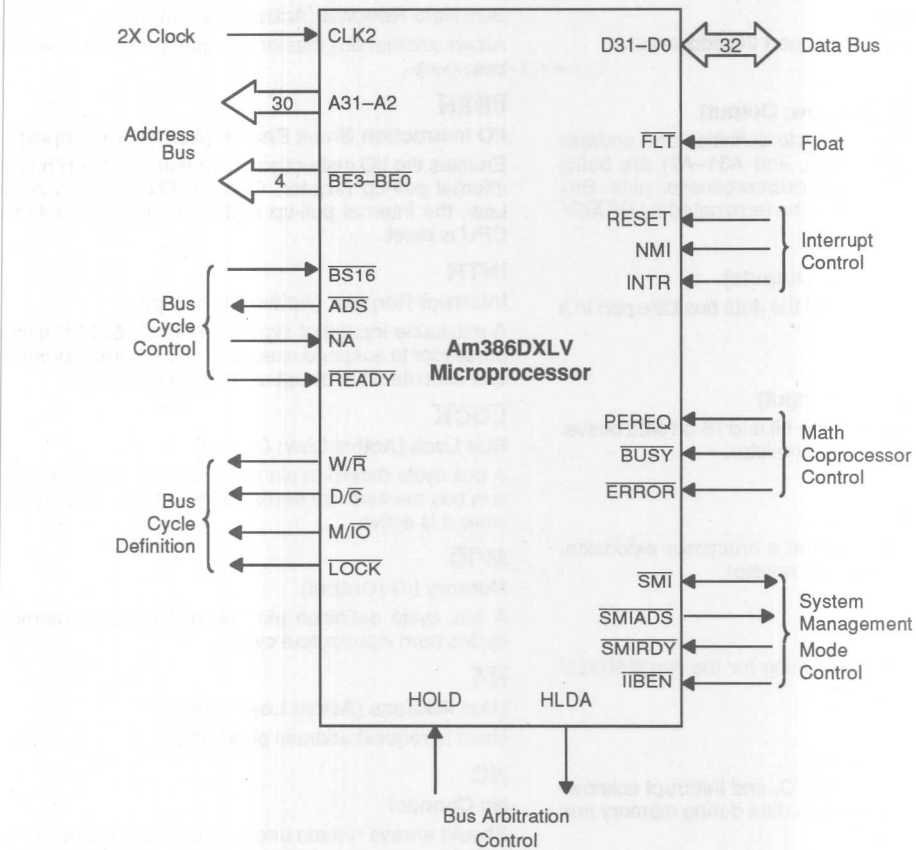


| Valid Combinations |           |      |
|--------------------|-----------|------|
| NG                 | 80386DXLV | -25  |
|                    |           | -25F |
|                    |           | -25S |
|                    |           |      |

**Valid Combinations**

Valid Combinations lists configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

## LOGIC SYMBOL



15021B-003

## PIN DESCRIPTION

### A31-A2

#### Address Bus (Outputs)

Outputs physical memory or port I/O addresses.

### ADS

#### Address Status (Active Low; Output)

Indicates that a valid bus cycle definition and address (W/R, D/C, M/I $\overline{O}$ , BE3-BE0, and A31-A2) are being driven at the Am386DXLV microprocessor pins. Bus cycles initiated by ADS must be terminated by READY.

### BE3-BE0

#### Byte Enable (Active Low; Outputs)

Indicates which data bytes of the data bus take part in a bus cycle.

### BS16

#### Bus Size 16 (Active Low; Input)

Allows direct connection of 32-bit and 16-bit data buses. BS16 has an internal pull-up resistor.

### BUSY

#### Busy (Active Low; Input)

Signals a busy condition from a processor extension. BUSY has an internal pull-up resistor.

### CLK2

#### Clock (Input)

Provides the fundamental timing for the Am386DXLV microprocessor.

### D31-D0

#### Data Bus (Inputs/Outputs)

Inputs data during memory, I/O, and interrupt acknowledge read cycles and outputs data during memory and I/O write cycles.

### D/C

#### Data/Control (Output)

A bus cycle definition pin that distinguishes data cycles, either memory or I/O, from control cycles (which are interrupt acknowledge, halt, and instruction fetching).

### ERROR

#### Error (Active Low; Input)

Signals an error condition from a processor extension. ERROR has an internal pull-up resistor.

### FLT

#### Float (Active Low; Input)

An input signal which forces all bidirectional and output signals, including HLDA, to the three-state condition. FLT has an internal pull-up resistor.

### HLDA

#### Bus Hold Acknowledge (Active High; Output)

Indicates that the Am386DXLV microprocessor has surrendered control of its local bus to another bus master.

### HOLD

#### Bus Hold Request (Active High; Input)

Allows another bus master to request control of the local bus.

### IIBEN

#### I/O Instruction Break Enable (Active Low; Input)

Enables the I/O instruction break feature. IIBEN has an internal pull-up resistor. Once IIBEN is driven active Low, the internal pull-up resistor is disabled until the CPU is reset.

### INTR

#### Interrupt Request (Active High; Input)

A maskable input that signals the Am386DXLV microprocessor to suspend execution of the current program and execute an interrupt acknowledge function.

### LOCK

#### Bus Lock (Active Low; Output)

A bus cycle definition pin that indicates that other system bus masters are denied access to the system bus while it is active.

### M/I $\overline{O}$

#### Memory I/O (Output)

A bus cycle definition pin that distinguishes memory cycles from input/output cycles.

### NA

#### Next Address (Active Low; Input)

Used to request address pipelining.

### NC

#### No Connect

Should always remain unconnected. Connection of an NC pin may cause the processor to malfunction or be incompatible with future steppings of the Am386DXLV microprocessor.

### NMI

#### Non-Maskable Interrupt Request (Active High; Input)

A non-maskable input that signals the Am386DXLV microprocessor to suspend execution of the current program and execute an interrupt acknowledge function.

### PEREQ

#### Processor Extension Request (Active High; Input)

Indicates that the processor extension has data to be transferred by the Am386DXLV microprocessor. PEREQ has an internal pull-down resistor.

### READY

#### Bus Ready (Active Low; Input)

Terminates the bus cycle initiated by ADS.

**RESET****Reset (Active High; Input)**

Suspends any operation in progress and places the Am386DXLV microprocessor in a known reset state.

**SMI****System Management Interrupt (Active Low; Input/Output)**

A Non-Maskable Interrupt (NMI) pin that signals the Am386DXLV microprocessor to suspend execution and enter System Management Mode. SMI has a dynamic pull-up resistor that is disabled when the processor is in SMM. SMI is not three-stated during hold acknowledge bus cycles.

**SMIADS****SMI Address Status (Active Low, Three-State; Output)**

Indicates that a valid bus cycle definition and address (W/R, D/C, M/I $\bar{O}$ , BE3-BE0, and A31-A2) are being driven at the Am386DXLV microprocessor pins while in System Management Mode. Bus cycles initiated by SMIADS must be terminated by SMIRDY.

**SMIRDY****SMI Ready (Active Low; Input)**

This input terminates the current bus cycle to the SMM address space in the same manner the  $\overline{READY}$  pin does for the Normal Mode address space.  $\overline{SMIRDY}$  has an internal pull-up resistor.  $\overline{SMIRDY}$  must not be tied to  $\overline{READY}$ .

**V<sub>cc</sub>****System Power (Active High; Input)**

Provides the DC supply input.

**V<sub>ss</sub>****System Ground (Input)**

Provides 0-V connection from which all inputs and outputs are measured.

**W/ $\bar{R}$** **Write/Read (Output)**

A bus cycle definition pin that distinguishes write cycles from read cycles.

# ABSOLUTE MAXIMUM RATINGS

Storage Temperature . . . . .  $-65^{\circ}\text{C}$  to  $+150^{\circ}\text{C}$   
Ambient Temperature Under Bias . .  $-65^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$

Stresses above those listed may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to ABSOLUTE MAXIMUM RATINGS for extended periods may affect device reliability.

# OPERATING RANGES

Supply Voltage with Respect to  $V_{SS}$  . .  $-0.5\text{ V}$  to  $+7\text{ V}$   
Voltage on Other Pins . . . . .  $-0.5\text{ V}$  to  $V_{CC} + 0.5\text{ V}$

Operating ranges define those limits between which the functionality of the device is guaranteed.

# DC CHARACTERISTICS over COMMERCIAL Operating Ranges

$V_{CC} = 3.0\text{ V}$  to  $3.6\text{ V}$ ;  $T_{CASE} = 0^{\circ}\text{C}$  to  $+100^{\circ}\text{C}$

| Symbol     | Parameter Description  | Notes  | Preliminary     |                         | Unit          |
|------------|--|--|-----------------|-------------------------|---------------|
|            |  |  | Min             | Max                     |               |
| $V_{IL}$   | Input Low Voltage  | (Note 1)   | $-0.3$          | $0.8$                   | V             |
| $V_{IH}$   | Input High Voltage   |  | $2.0$           | $V_{CC} + 0.3$          | V             |
| $V_{ILC}$  | CLK2 Input Low Voltage   | (Note 1)   | $-0.3$          | $0.8$                   | V             |
| $V_{IHC}$  | CLK2 Input High Voltage (25 MHz)   |  | $2.4$           | $V_{CC} + 0.3$          | V             |
| $V_{OL}$   | Output Low Voltage   | (Note 5)   |                 |                         |               |
|            | $I_{OL} = 0.5\text{ mA}$ : A31–A2, D31–D0  |  |                 | $0.2$                   | V             |
|            | $I_{OL} = 0.5\text{ mA}$ : $\overline{BE3}$ – $\overline{BE0}$ , W/R, D/C, M/I $\overline{O}$ , LOCK, ADS, SMIADS, HLDA, SMI |  |                 | $0.2$                   | V             |
|            | $I_{OL} = 2\text{ mA}$ : A31–A2, D31–D0  |  |                 | $0.45$                  | V             |
| $V_{OH}$   | Output High Voltage  | (Note 5)<br>(Note 6)   |                 |                         |               |
|            | $I_{OH} = 0.1\text{ mA}$ : A31–A2, D31–D0  |  | $V_{CC} - 0.2$  |                         | V             |
|            | $I_{OH} = 0.1\text{ mA}$ : $\overline{BE3}$ – $\overline{BE0}$ , W/R, D/C, M/I $\overline{O}$ , LOCK, ADS, SMIADS, HLDA, SMI |  | $V_{CC} - 0.2$  |                         | V             |
|            | $I_{OH} = 0.5\text{ mA}$ : A31–A2, D31–D0  |  | $V_{CC} - 0.45$ |                         | V             |
| $I_{LI}$   | Input Leakage Current (All pins except $\overline{BS16}$ , PEREQ, $\overline{IIBEN}$ , BUSY, FLT, ERROR, SMI, and SMIRDY)    | $0\text{ V} \leq V_{IN} \leq V_{CC}$<br>(Note 7)                     |                 | $\pm 10$                | $\mu\text{A}$ |
|            |  |  |                 |                         |               |
| $I_{IH}$   | Input Leakage Current (PEREQ Pin)  | $V_{IH} = V_{CC} - 0.1\text{ V}$<br>$V_{IH} = 2.4\text{ V}$ (Note 2) |                 | $300$<br>$200$          | $\mu\text{A}$ |
| $I_{IL}$   | Input Leakage Current ( $\overline{BS16}$ , BUSY, FLT, ERROR, SMI, $\overline{IIBEN}$ , and SMIRDY)                          | $V_{IL} = 0.1\text{ V}$<br>$V_{IL} = 0.45\text{ V}$ (Note 3)         |                 | $-300$<br>$-200$        | $\mu\text{A}$ |
| $I_{LO}$   | Output Leakage Current   | $0.1\text{ V} \leq V_{OUT} \leq V_{CC}$                              |                 | $\pm 15$                | $\mu\text{A}$ |
| $I_{CC}$   | Supply Current (Note 8)  | $V_{CC} = 3.3\text{ V}$  |                 | $V_{CC} = 3.6\text{ V}$ |               |
|            | CLK2 = 40 MHz: Oper. Freq. 20 MHz  | $I_{CC}\text{ Typ} = 80$   |                 | $95$                    | mA            |
|            | CLK2 = 50 MHz: Oper. Freq. 25 MHz  | $I_{CC}\text{ Typ} = 95$   |                 | $115$                   | mA            |
| $I_{CCSB}$ | Standby Current (Note 8)   | $I_{CCSB}\text{ Typ} = 10\text{ }\mu\text{A}$                        |                 | $150$                   | $\mu\text{A}$ |
| $C_{IN}$   | Input or I/O Capacitance   | $F_C = 1\text{ MHz}$ (Note 4)  |                 | $10$                    | pF            |
| $C_{OUT}$  | Output Capacitance   | $F_C = 1\text{ MHz}$ (Note 4)  |                 | $12$                    | pF            |
| $C_{CLK}$  | CLK2 Capacitance   | $F_C = 1\text{ MHz}$ (Note 4)  |                 | $20$                    | pF            |

- Notes: 1. The Min value,  $-0.3$ , is not 100% tested.  
2. PEREQ input has an internal pull-down resistor.  
3.  $\overline{BS16}$ , BUSY, FLT, ERROR, SMI, SMIRDY, and  $\overline{IIBEN}$  inputs each have an internal pull-up resistor.  
4. Not 100% tested.  
5. Outputs are CMOS and will pull rail-to-rail if the load is not resistive.  
6.  $V_{OH}$  is only valid for SMI when exiting SMM for two CLK2 cycles.  
7. SMI and  $\overline{IIBEN}$  leakage Low will be  $I_{LI}$  when pull-up is inactive and  $I_{IL}$  when pull-up is active.  
8. Inputs are at either  $V_{CC}$  or  $V_{SS}$ .



**ABSOLUTE MAXIMUM RATINGS**

Storage Temperature . . . . . -65°C to +150°C  
 Ambient Temperature Under Bias . . -65°C to +125°C

*Stresses above those listed may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to ABSOLUTE MAXIMUM RATINGS for extended periods may affect device reliability.*

**OPERATING RANGES**

Supply Voltage with Respect to  $V_{SS}$  . . -0.5 V to +7 V  
 Voltage on Other Pins . . . . . -0.5 V to  $V_{CC} + 0.5$  V

*Operating ranges define those limits between which the functionality of the device is guaranteed.*

**DC CHARACTERISTICS over COMMERCIAL Operating Ranges**

$V_{CC} = 4.5$  V to 5.5 V;  $T_{CASE} = 0^\circ\text{C}$  to +100°C

| Symbol     | Parameter Description   | Notes   | Preliminary |                  | Unit          |
|------------|---|---|-------------|------------------|---------------|
|            |   |   | Min         | Max              |               |
| $V_{IL}$   | Input Low Voltage   | (Note 1)  | -0.3        | 0.8              | V             |
| $V_{IH}$   | Input High Voltage  |   | 2.0         | $V_{CC} + 0.3$   | V             |
| $V_{ILC}$  | CLK2 Input Low Voltage  | (Note 1)  | -0.3        | 0.8              | V             |
| $V_{IHC}$  | CLK2 Input High Voltage (33 MHz)  |   | 2.7         | $V_{CC} + 0.3$   | V             |
| $V_{OL}$   | Output Low Voltage<br>$I_{OL} = 4$ mA: A31-A2, D31-D0<br>$I_{OL} = 5$ mA: BE3-BE0, W/R, D/C, M/I/O, LOCK, ADS, SMIADS, HLDA, SMI    | (Note 5)  |             | 0.45             | V             |
|            |   |   |             | 0.45             | V             |
| $V_{OH}$   | Output High Voltage<br>$I_{OH} = 1$ mA: A31-A2, D31-D0<br>$I_{OH} = 0.9$ mA: BE3-BE0, W/R, D/C, M/I/O, LOCK, ADS, SMIADS, HLDA, SMI | (Note 5)<br>(Note 6)                              | 2.4<br>2.4  |                  | V<br>V        |
|            |   |   |             |                  |               |
| $I_{LI}$   | Input Leakage Current<br>(All pins except BS16, PEREQ, IIBEN, BUSY, FLT, SMI, SMIRDY, and ERROR)                                    | $0 \text{ V} \leq V_{IN} \leq V_{CC}$<br>(Note 7) |             | $\pm 15$         | $\mu\text{A}$ |
| $I_{IH}$   | Input Leakage Current<br>(PEREQ Pin)  | $V_{IH} = 2.4$ V<br>(Note 2)                      |             | 200              | $\mu\text{A}$ |
| $I_{IL}$   | Input Leakage Current<br>(BS16, BUSY, FLT, SMI, SMIRDY, IIBEN, and ERROR)   | $V_{IL} = 0.45$<br>(Note 3)                       |             | -400             | $\mu\text{A}$ |
| $I_{LO}$   | Output Leakage Current  | $0.45 \text{ V} \leq V_{OUT} \leq V_{CC}$         |             | $\pm 15$         | $\mu\text{A}$ |
| $I_{CC}$   | Supply Current (Note 8)   | $V_{CC} = 5.0$ V                                  |             | $V_{CC} = 5.5$ V |               |
|            | CLK2 = 40 MHz: Oper. Freq. 20 MHz   | $I_{CC}$ Typ = 130                                |             | 155              | mA            |
|            | CLK2 = 50 MHz: Oper. Freq. 25 MHz   | $I_{CC}$ Typ = 160                                |             | 190              | mA            |
|            | CLK2 = 66 MHz: Oper. Freq. 33 MHz   | $I_{CC}$ Typ = 210                                |             | 245              | mA            |
| $I_{CCSB}$ | Standby Current (Note 8)  | $I_{CCSB}$ Typ = 20 $\mu\text{A}$                 |             | 150              | $\mu\text{A}$ |
| $C_{IN}$   | Input or I/O Capacitance  | $F_C = 1$ MHz (Note 4)                            |             | 10               | pF            |
| $C_{OUT}$  | Output Capacitance  | $F_C = 1$ MHz (Note 4)                            |             | 12               | pF            |
| $C_{CLK}$  | CLK2 Capacitance  | $F_C = 1$ MHz (Note 4)                            |             | 20               | pF            |

Notes: 1. The Min value, -0.3, is not 100% tested.

2. PEREQ input has an internal pull-down resistor.

3. BS16, BUSY, FLT, ERROR, SMI, SMIRDY, and IIBEN inputs each have an internal pull-up resistor.

4. Not 100% tested.

5. Outputs are CMOS and will pull rail-to-rail if the load is not resistive.

6.  $V_{OH}$  is only valid for SMI when exiting SMM for two CLK2 cycles.

7. SMI and IIBEN leakage Low will be  $I_{LI}$  when pull-up is inactive and  $I_{IL}$  when pull-up is active.

8. Inputs are at either  $V_{CC}$  or  $V_{SS}$ .

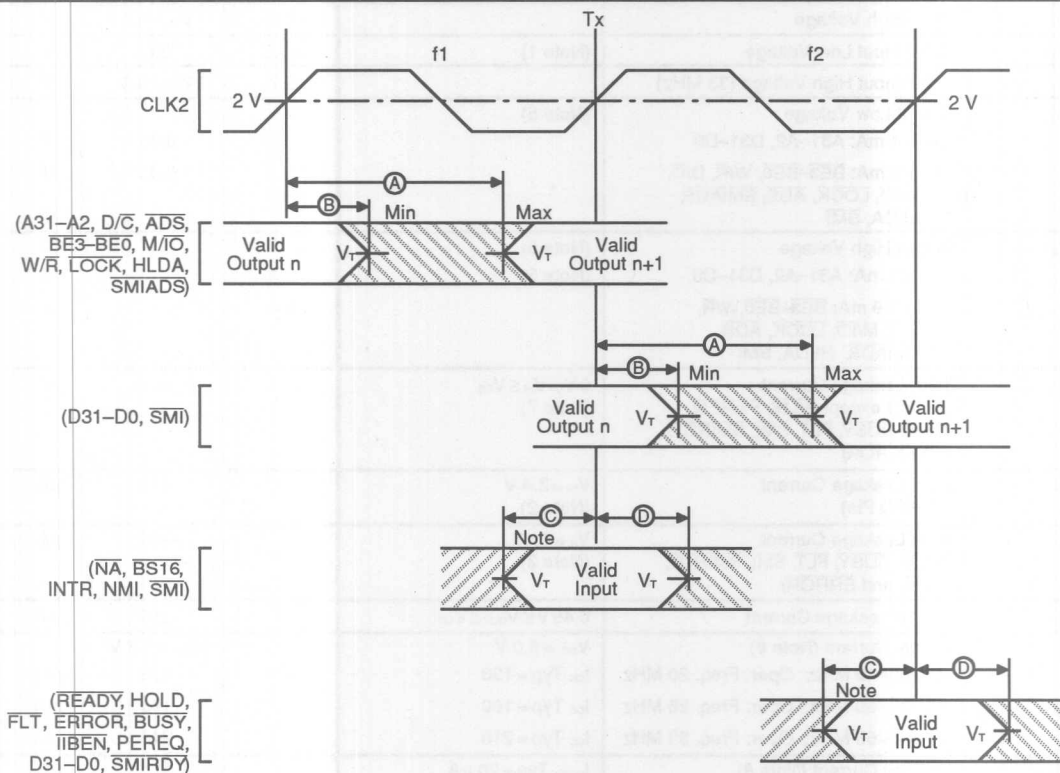
## SWITCHING CHARACTERISTICS

The switching characteristics consist of output delays, input setup requirements, and input hold requirements. All characteristics are relative to the CLK2 rising edge crossing the 2.0-V level.

Switching characteristic measurement is defined in Figure 2. Inputs must be driven to the voltage levels indicated by this diagram. Am386DXLV CPU output delays are specified with minimum and maximum limits measured as shown. The minimum Am386DXLV microprocessor delay times are hold times provided to external circuitry. Am386DXLV microprocessor input setup and hold time are specified as minimums, defining the smallest acceptable sampling window. Within the sampling

window, a synchronous input signal must be stable for correct Am386DXLV microprocessor operation.

Outputs W/R, D/C, M/I/O, LOCK, BE3-BE0, ADS, A31-A2, HLDA, and SMIADS only change at the beginning of phase one. D31-D0 (write cycles) and SMI only change at the beginning of phase two. The READY, HOLD, IIBEN, BUSY, ERROR, PEREQ, FLT, D31-D0, and SMIRDY (read cycles) inputs are sampled at the beginning of phase one. The NA, BS16, INTR, NMI, and SMI inputs are sampled at the beginning of phase two.



Legend: A—Maximum Output Delay Spec  
B—Minimum Output Delay Spec  
C—Minimum Input Setup Spec  
D—Minimum Input Hold Spec

Notes: 1. Input waveforms have  $t_r \leq 2.0$  ns from 0.8 V to 2.0 V.  
2.  $V_t = 1.0$  V at  $V_{cc} \leq 3.6$ ; 1.5 V at  $V_{cc} > 3.6$ .

16306B-003

Figure 2. Drive Levels and Measurement Points

**SWITCHING CHARACTERISTICS over operating range at 25 MHz** $V_{CC} = 3.0 \text{ V to } 3.6 \text{ V}$ ;  $T_{CASE} = 0^\circ\text{C to } +100^\circ\text{C}$ 

| No. | Parameter Description                     | Notes                 | Ref Figures | Preliminary |     | Unit |
|-----|---|-----------------------|-------------|-------------|-----|------|
|     |   |                       |             | Min         | Max |      |
|     | Operating Frequency                       | Half of CLK2 Freq     |             | 0           | 25  | MHz  |
| 1   | CLK2 Period                               |                       | 4           | 20          |     | ns   |
| 2   | CLK2 High Time                            | at $V_{IH}$           | 4           | 4           |     | ns   |
| 3   | CLK2 Low Time                             | at 0.8 V              | 4           | 5           |     | ns   |
| 4   | CLK2 Fall Time (Note 3)                   | 2.4 V to 0.8 V        | 4           |             | 7   | ns   |
| 5   | CLK2 Rise Time (Note 3)                   | 0.8 V to 2.4 V        | 4           |             | 7   | ns   |
| 6   | A31–A2 Valid Delay                        | $C_L = 50 \text{ pF}$ | 3, 6        | 4           | 17  | ns   |
| 7   | A31–A2 Float Delay                        | (Note 1)              | 13          | 4           | 30  | ns   |
| 8   | BE3–BE0, LOCK Valid Delay                 | $C_L = 50 \text{ pF}$ | 3, 6        | 4           | 17  | ns   |
| 9   | BE3–BE0, LOCK Float Delay                 | (Note 1)              | 13          | 4           | 30  | ns   |
| 10  | W/R, M/IO, D/C, ADS Valid Delay           | $C_L = 50 \text{ pF}$ | 3, 6        | 4           | 17  | ns   |
| 10s | SMIADS Valid Delay                        | $C_L = 50 \text{ pF}$ | 3, 6        | 4           | 25  | ns   |
| 11  | W/R, M/IO, D/C, ADS Float Delay           | (Note 1)              | 13          | 4           | 30  | ns   |
| 11s | SMIADS Float Delay                        | (Note 1)              | 13          | 4           | 30  | ns   |
| 12  | D31–D0 Write Data Valid Delay             | $C_L = 50 \text{ pF}$ | 6, 7        | 7           | 23  | ns   |
| 12a | D31–D0 Write Data Hold Time               | $C_L = 50 \text{ pF}$ | 3, 8        | 2           |     | ns   |
| 13  | D31–D0 Float Delay                        | (Note 1)              | 13          | 4           | 22  | ns   |
| 14  | HLDA Valid Delay                          | $C_L = 50 \text{ pF}$ | 3, 13       | 4           | 22  | ns   |
| 14f | HLDA Float Delay                          | (Note 1)              | 14          | 4           | 30  | ns   |
| 15  | NA Setup Time                             |                       | 5           | 5           |     | ns   |
| 16  | NA Hold Time                              |                       | 5           | 3           |     | ns   |
| 17  | BS16 Setup Time                           |                       | 5           | 5           |     | ns   |
| 18  | BS16 Hold Time                            |                       | 5           | 3           |     | ns   |
| 19  | READY Setup Time                          |                       | 5           | 9           |     | ns   |
| 19s | SMIRDY Setup Time                         |                       | 5           | 9           |     | ns   |
| 20  | READY Hold Time                           |                       | 5           | 4           |     | ns   |
| 20s | SMIRDY Hold Time                          |                       | 5           | 4           |     | ns   |
| 21  | D31–D0 Read Setup Time                    |                       | 5           | 7           |     | ns   |
| 22  | D31–D0 Read Hold Time                     |                       | 5           | 5           |     | ns   |
| 23  | HOLD Setup Time                           |                       | 5           | 9           |     | ns   |
| 24  | HOLD Hold Time                            |                       | 5           | 3           |     | ns   |
| 25  | RESET Setup Time                          |                       | 15          | 8           |     | ns   |
| 26  | RESET Hold Time                           |                       | 15          | 3           |     | ns   |
| 27  | NMI, INTR Setup Time                      | (Note 2)              | 5           | 6           |     | ns   |
| 27s | SMI Setup Time                            |                       | 5           | 6           |     | ns   |
| 28  | NMI, INTR Hold Time                       | (Note 2)              | 5           | 6           |     | ns   |
| 28s | SMI Hold Time                             |                       | 5           | 4           |     | ns   |
| 29  | PEREQ, ERROR, BUSY, FLT, IIBEN Setup Time | (Note 2)              | 5           | 6           |     | ns   |
| 30  | PEREQ, ERROR, BUSY, FLT, IIBEN Hold Time  | (Note 2)              | 5           | 5           |     | ns   |
| 31  | SMI Valid Delay                           |                       | 5, 13       | 4           | 22  | ns   |
| 32  | SMI Float Delay                           | (Note 1)              | 14          | 4           | 30  | ns   |

Notes: 1. Float condition occurs when maximum output current becomes less than  $I_{LO}$  magnitude. Float delay is not 100% tested.

2. These inputs are allowed to be asynchronous to CLK2. The setup and hold specifications are given for testing purposes, to assure recognition within a specific CLK2 period.

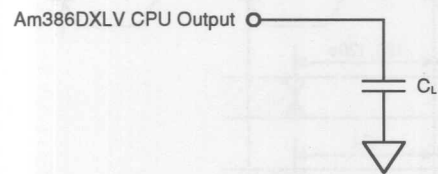
3. Rise and fall times are not tested.

**SWITCHING CHARACTERISTICS over operating range at 33 MHz**
 $V_{CC} = 4.5 \text{ V to } 5.5 \text{ V}$ ;  $T_{CASE} = 0^{\circ}\text{C to } +100^{\circ}\text{C}$ 

| No. | Parameter Description  | Notes                 | Ref Figures | Preliminary |      | Unit |
|-----|--|-----------------------|-------------|-------------|------|------|
|     |  |                       |             | Min         | Max  |      |
|     | Operating Frequency  | Half of CLK2 Freq     |             | 0           | 33.3 | MHz  |
| 1   | CLK2 Period  |                       | 4           | 15.0        |      | ns   |
| 2   | CLK2 High Time   | at $V_{IHC}$          | 4           | 4.5         |      | ns   |
| 3   | CLK2 Low Time  | at 0.8 V              | 4           | 4.5         |      | ns   |
| 4   | CLK2 Fall Time (Note 3)                                      | 2.4 V to 0.8 V        | 4           |             | 4    | ns   |
| 5   | CLK2 Rise Time (Note 3)                                      | 0.8 V to 2.4 V        | 4           |             | 4    | ns   |
| 6   | A31–A2 Valid Delay   | $C_L = 50 \text{ pF}$ | 3, 6        | 4           | 15   | ns   |
| 7   | A31–A2 Float Delay   | (Note 1)              | 13          | 4           | 20   | ns   |
| 8   | BE3–BE0, LOCK Valid Delay                                    | $C_L = 50 \text{ pF}$ | 3, 6        | 4           | 15   | ns   |
| 9   | BE3–BE0, LOCK Float Delay                                    | (Note 1)              | 13          | 4           | 20   | ns   |
| 10  | W/R, M/ $\overline{O}$ , D/ $\overline{C}$ , ADS Valid Delay | $C_L = 50 \text{ pF}$ | 3, 6        | 4           | 15   | ns   |
| 10s | SMIADS Valid Delay   | $C_L = 50 \text{ pF}$ | 2, 6        | 4           | 15   | ns   |
| 11  | W/R, M/ $\overline{O}$ , D/ $\overline{C}$ , ADS Float Delay | (Note 1)              | 13          | 4           | 20   | ns   |
| 11s | SMIADS Float Delay   | (Note 1)              | 13          | 4           | 20   | ns   |
| 12  | D31–D0 Write Data Valid Delay                                | $C_L = 50 \text{ pF}$ | 6, 7        | 7           | 23   | ns   |
| 12a | D31–D0 Write Data Hold Time                                  | $C_L = 50 \text{ pF}$ | 3, 8        | 2           |      | ns   |
| 13  | D31–D0 Float Delay   | (Note 1)              | 13          | 4           | 17   | ns   |
| 14  | HLDA Valid Delay   | $C_L = 50 \text{ pF}$ | 3, 13       | 4           | 20   | ns   |
| 14f | HLDA Float Delay   | (Note 1)              | 14          | 4           | 20   | ns   |
| 15  | NA Setup Time  |                       | 5           | 5           |      | ns   |
| 16  | NA Hold Time   |                       | 5           | 2           |      | ns   |
| 17  | BS16 Setup Time  |                       | 5           | 5           |      | ns   |
| 18  | BS16 Hold Time   |                       | 5           | 2           |      | ns   |
| 19  | READY Setup Time   |                       | 5           | 7           |      | ns   |
| 19s | SMIRDY Setup Time  |                       | 5           | 7           |      | ns   |
| 20  | READY Hold Time  |                       | 5           | 4           |      | ns   |
| 20s | SMIRDY Hold Time   |                       | 5           | 4           |      | ns   |
| 21  | D31–D0 Read Setup Time                                       |                       | 5           | 5           |      | ns   |
| 22  | D31–D0 Read Hold Time  |                       | 5           | 3           |      | ns   |
| 23  | HOLD Setup Time  |                       | 5           | 9           |      | ns   |
| 24  | HOLD Hold Time   |                       | 5           | 2           |      | ns   |
| 25  | RESET Setup Time   |                       | 15          | 5           |      | ns   |
| 26  | RESET Hold Time  |                       | 15          | 2           |      | ns   |
| 27  | NMI, INTR Setup Time   | (Note 2)              | 5           | 5           |      | ns   |
| 27s | SMI Setup Time   |                       | 5           | 5           |      | ns   |
| 28  | NMI, INTR Hold Time  | (Note 2)              | 5           | 5           |      | ns   |
| 28s | SMI Hold Time  |                       | 5           | 4           |      | ns   |
| 29  | PEREQ, ERROR, BUSY, FLT, IIBEN Setup Time                    | (Note 2)              | 5           | 5           |      | ns   |
| 30  | PEREQ, ERROR, BUSY, FLT, IIBEN Hold Time                     | (Note 2)              | 5           | 4           |      | ns   |
| 31  | SMI Valid Delay  |                       | 5, 13       | 4           | 17   | ns   |
| 32  | SMI Float Delay  | (Note 1)              | 14          | 4           | 20   | ns   |

- Notes: 1. Float condition occurs when maximum output current becomes less than  $I_{LO}$  in magnitude. Float delay is not 100% tested.  
2. These inputs are allowed to be asynchronous to CLK2. The setup and hold specifications are given for testing purposes, to assure recognition within a specific CLK2 period.  
3. Rise and fall times are not tested.

## SWITCHING CHARACTERISTICS (continued)



$C_L$  includes all parasitic capacitances.

Figure 3. AC Test Load

15021B-072

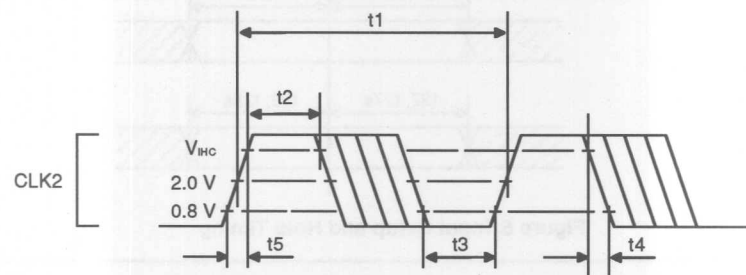
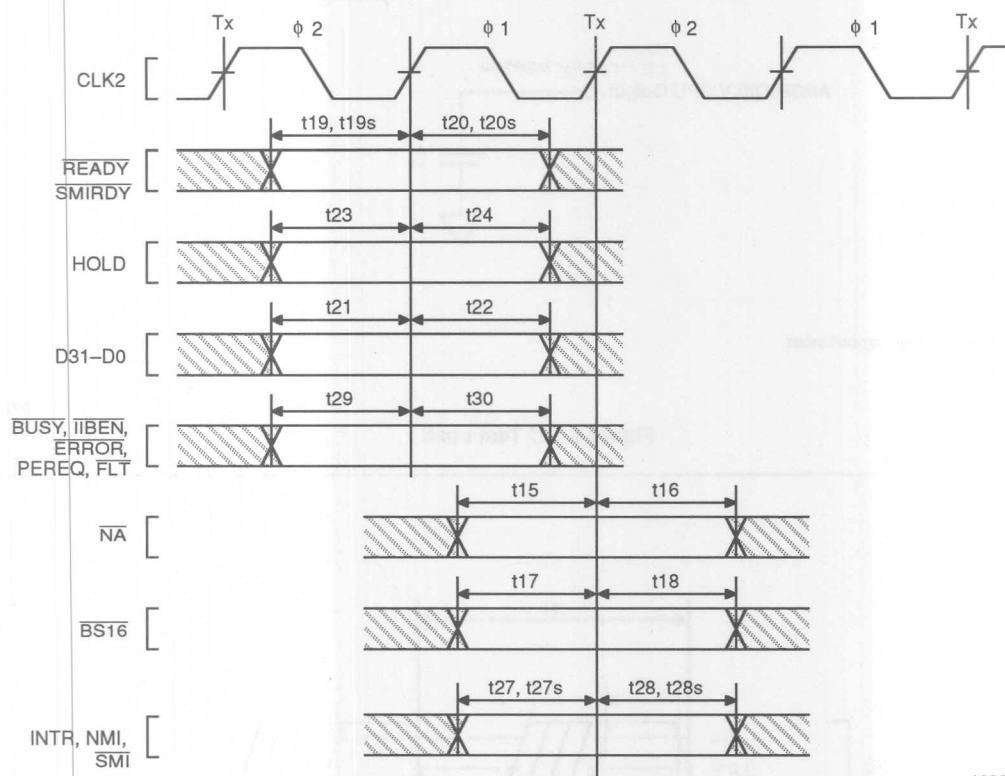


Figure 4. CLK2 Timing

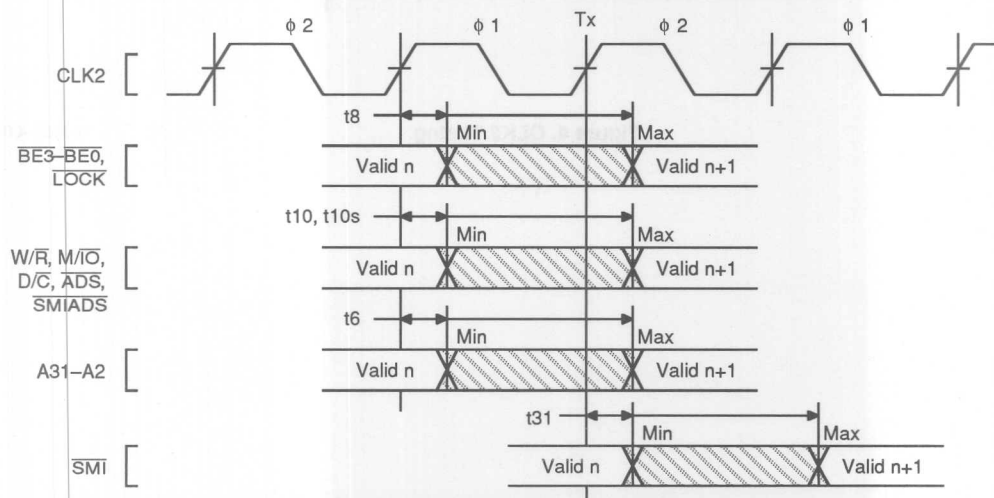
16306B-004

# SWITCHING WAVEFORMS



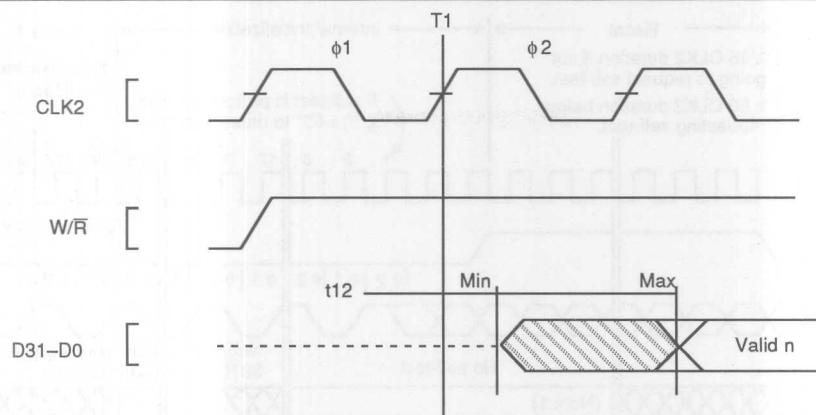
16306B-005

Figure 5. Input Setup and Hold Timing



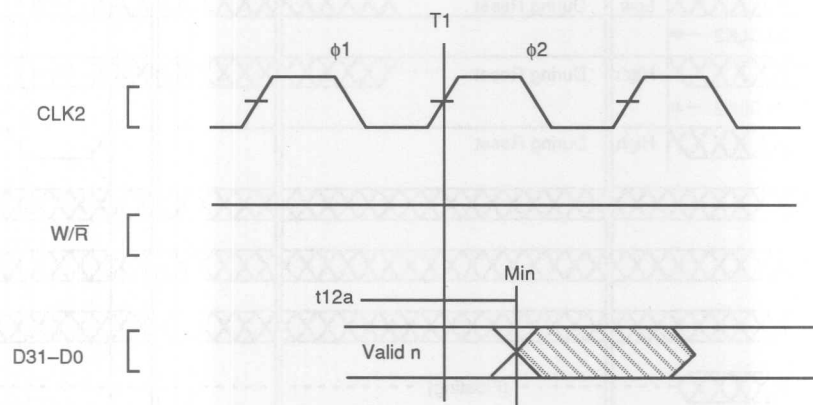
16306B-006

Figure 6. Output Valid Delay Timing



15021B-076

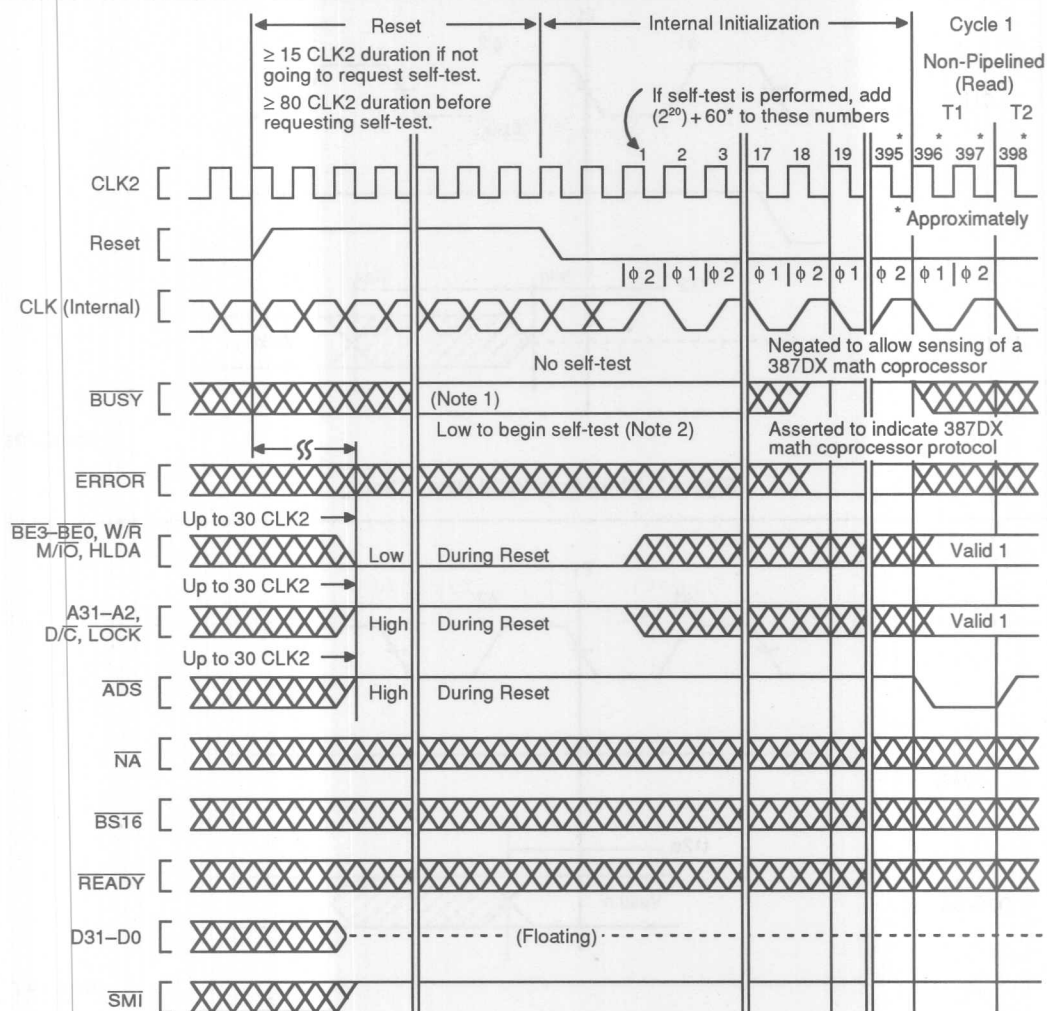
Figure 7. Write Data Valid Delay Timing



15021B-077

Figure 8. Write Data Hold Timing





Notes: 1. BUSY should be held stable for eight CLK2 periods before and after the CLK2 period in which RESET falling edge occurs.  
2. If self-test is requested, the Am386DXLV microprocessor outputs remain in their reset state as shown here.

16306B-007

Figure 9. Bus Activity from Reset Until First Code Fetch

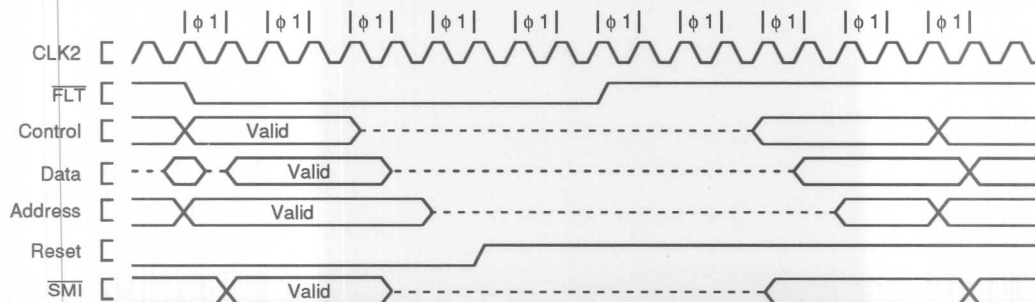
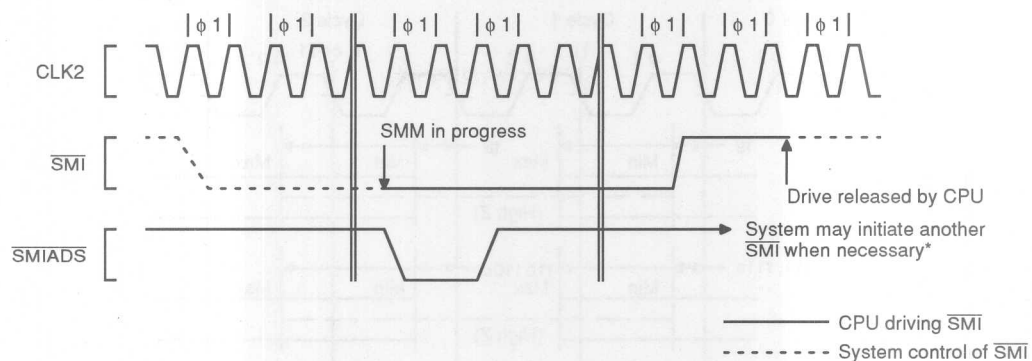


Figure 10. Entering and Exiting FLT

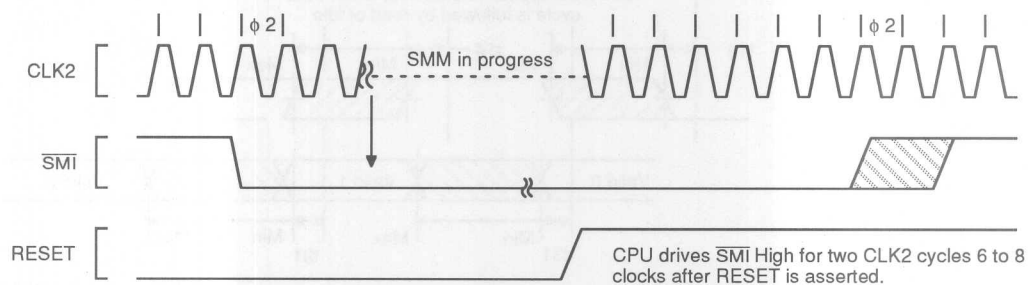
16306B-008



\*Once initiated, the system must hold  $\overline{\text{SMI}}$  Low until the first  $\overline{\text{SMIADS}}$ . At this time, the system releases control of  $\overline{\text{SMI}}$  and can not drive  $\overline{\text{SMI}}$  until three CLK2 cycles after the CPU drives  $\overline{\text{SMI}}$  High. CPU will drive  $\overline{\text{SMI}}$  High for two CLK2 cycles. The additional clock allows the CPU to completely release  $\overline{\text{SMI}}$  and prevents any driver overlap.

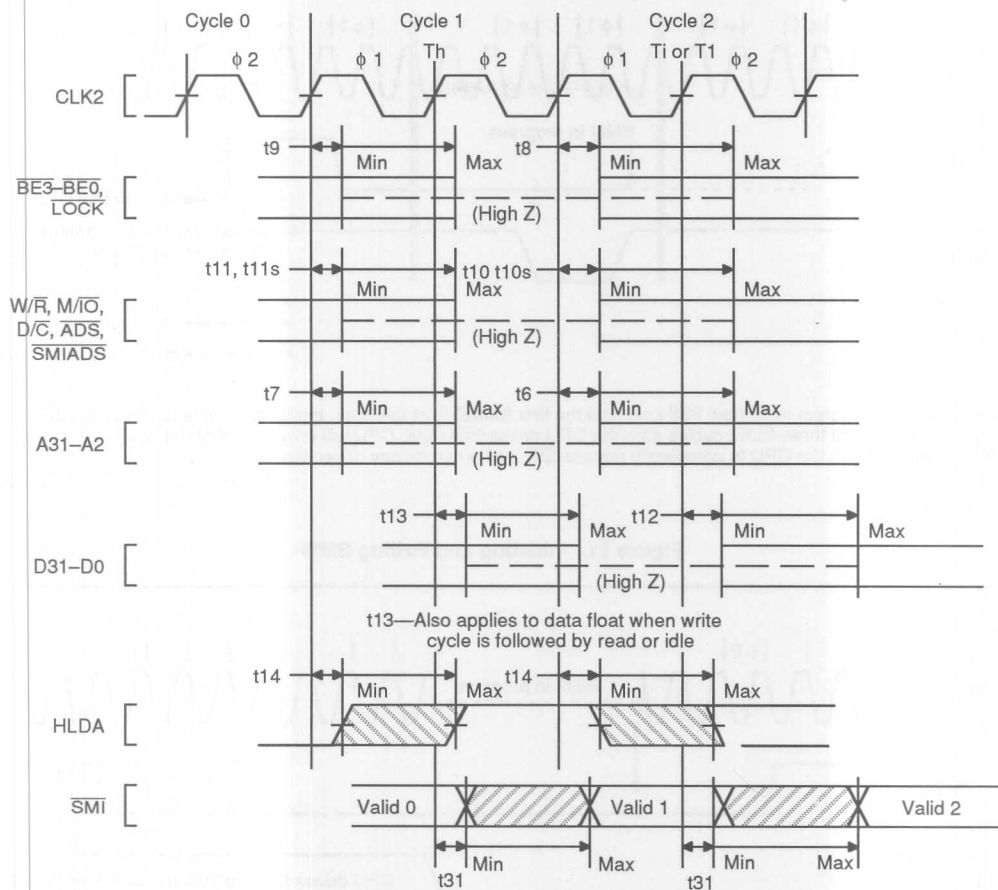
16306B-009

Figure 11. Initiating and Exiting SMM



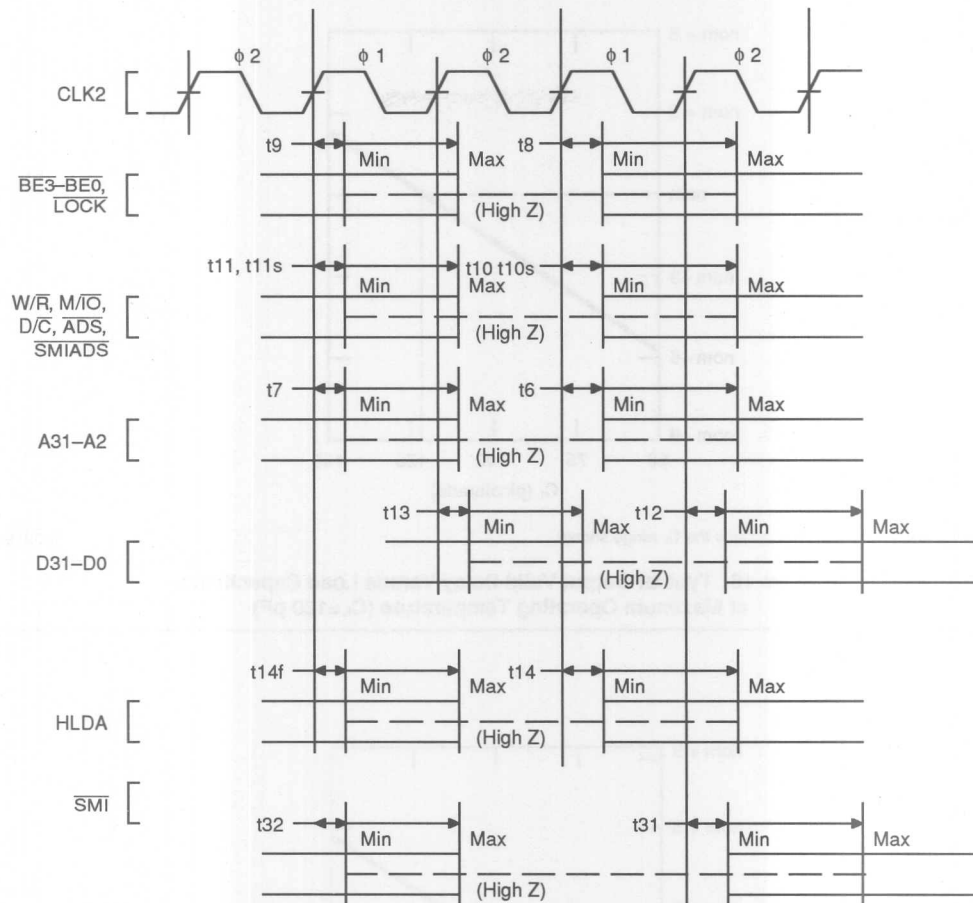
16306B-010

Figure 12. RESET and  $\overline{\text{SMI}}$

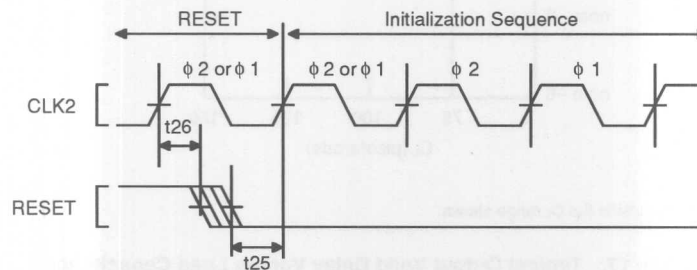


16306B-011

Figure 13. Output Float Delay and HLDA and  $\overline{\text{SMI}}$  Valid Delay Timing

Figure 14. Output Float Delay Entering and Exiting  $\overline{\text{FLT}}$ 

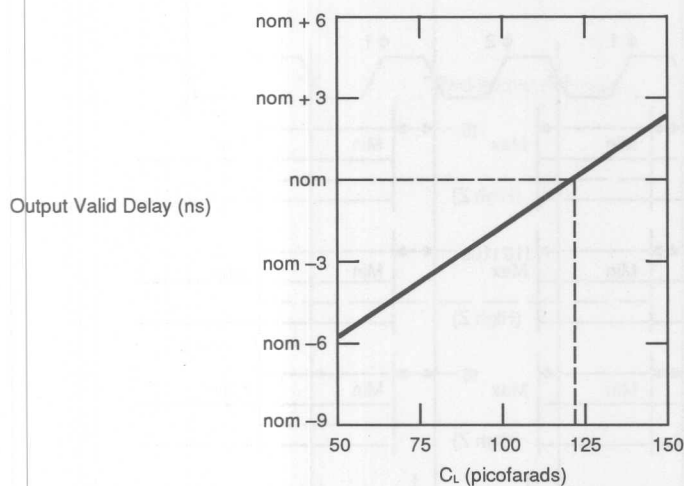
16306B-012



The second internal processor phase following RESET High-to-Low transition (provided  $t_{25}$  and  $t_{26}$  are met) is  $\phi 2$ .

15021B-084

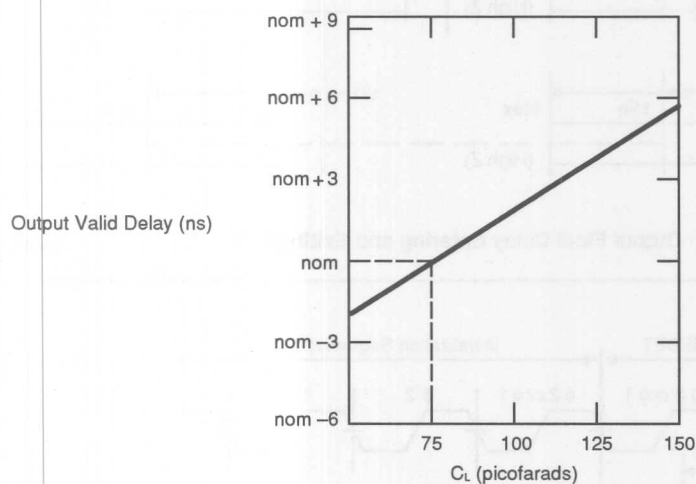
Figure 15. RESET Setup and Hold Timing and Internal Phase



Note: This graph will not be linear outside the  $C_L$  range shown.

15021B-079

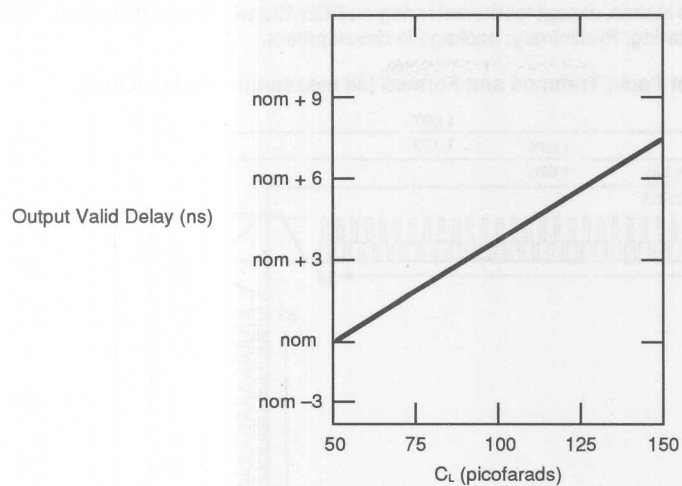
**Figure 16. Typical Output Valid Delay Versus Load Capacitance at Maximum Operating Temperature ( $C_L = 120$  pF)**



Note: This graph will not be linear outside the  $C_L$  range shown.

15021B-080

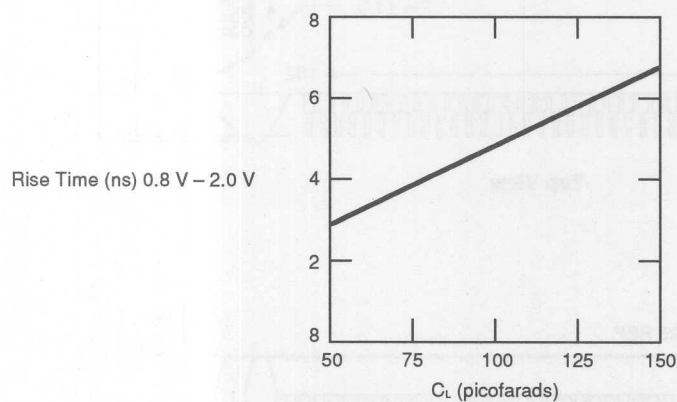
**Figure 17. Typical Output Valid Delay Versus Load Capacitance at Maximum Operating Temperature ( $C_L = 75$  pF)**



Note: This graph will not be linear outside the  $C_L$  range shown.

**Figure 18. Typical Output Valid Delay Versus Load Capacitance at Maximum Operating Temperature ( $C_L = 50$  pF)**

15021B-081



Note: This graph will not be linear outside the  $C_L$  range shown.

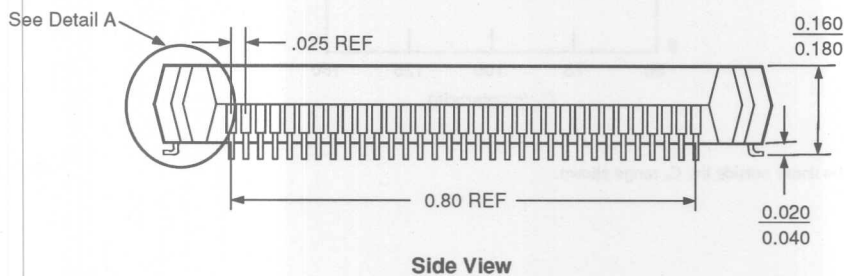
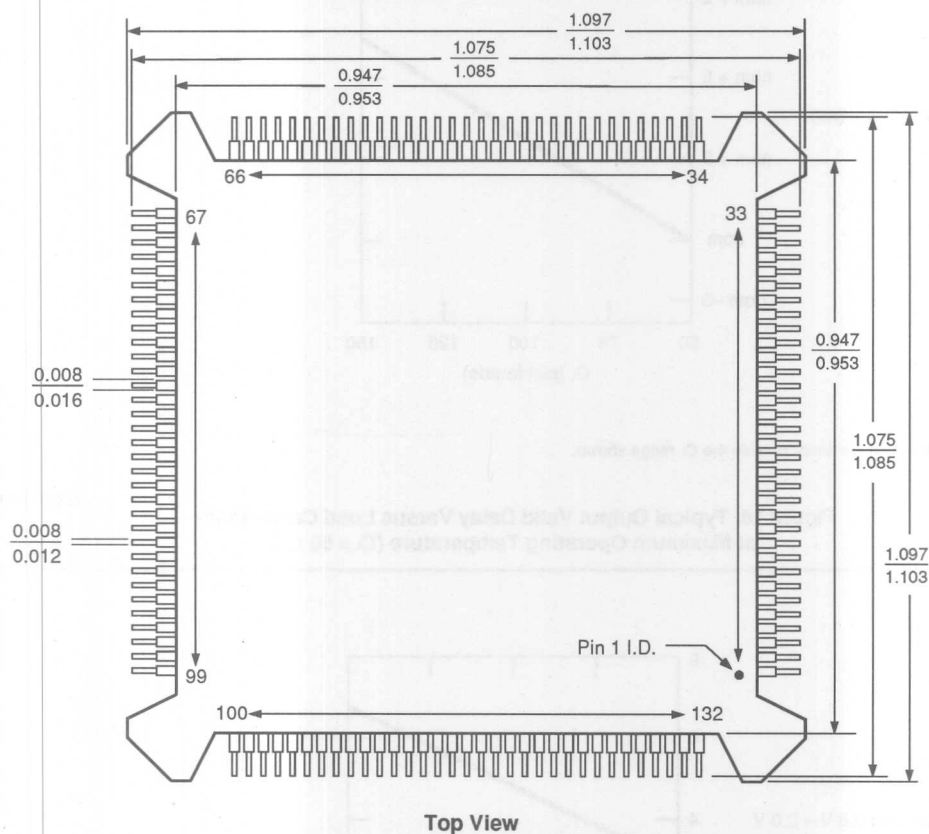
**Figure 19. Typical Output Rise Time Versus Load Capacitance at Maximum Operating Temperature**

15021B-082

# PHYSICAL DIMENSIONS

For reference only. All dimensions are in inches, except for the outer ring on PQB 132 which is in millimeters. BSC is an ANSI standard for Basic Space Centering. Preliminary; package in development.

PQ 132 — Plastic Quad Flat Pack; Trimmed and Formed (all measurements in inches)

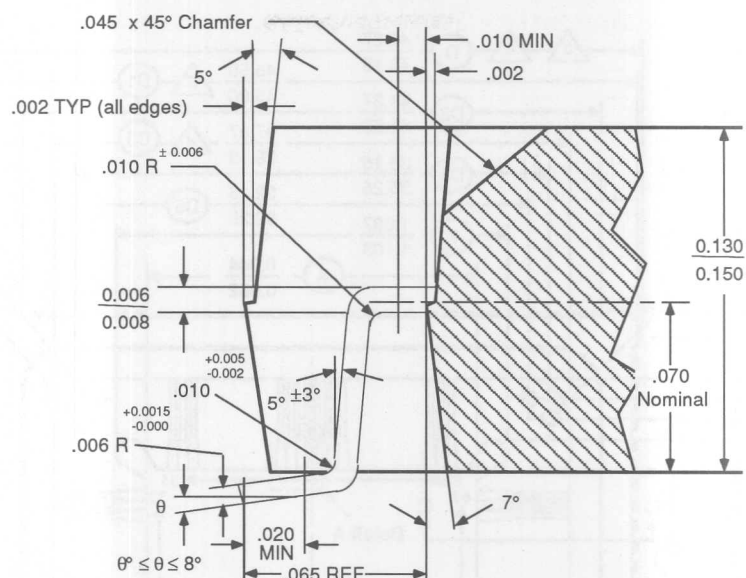


16423A  
BX 43  
4/22/92 SG



## PHYSICAL DIMENSIONS (continued)

## PQ 132 — Plastic Quad Flat Pack; Trimmed and Formed (continued)

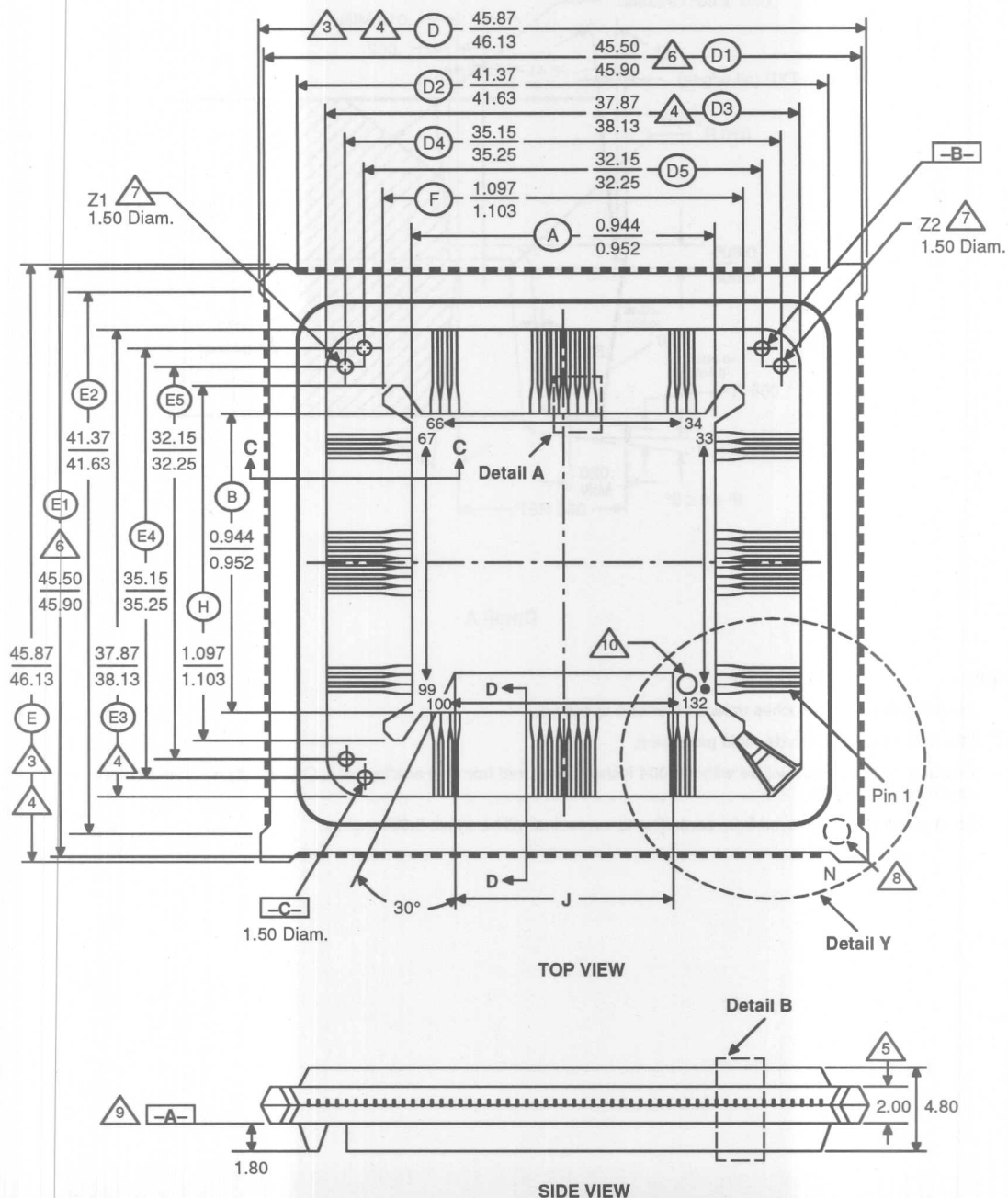


Detail A

## Notes:

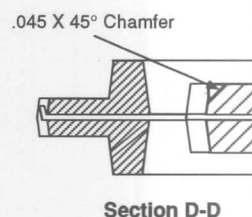
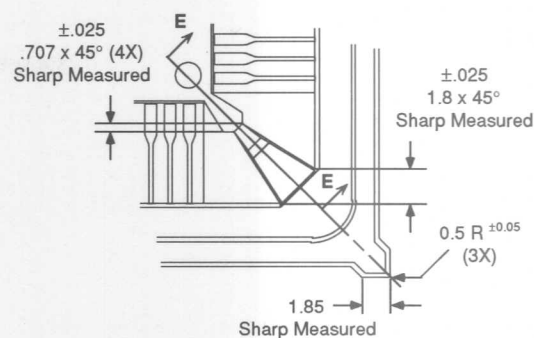
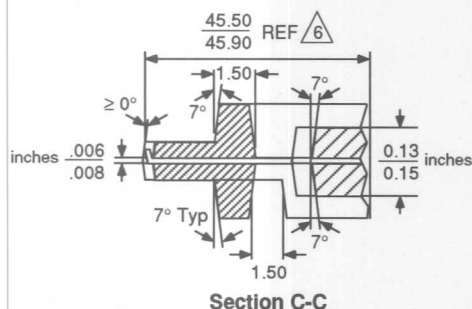
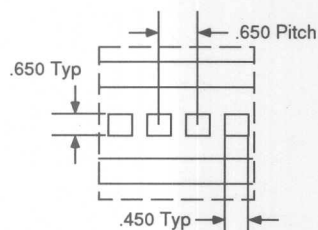
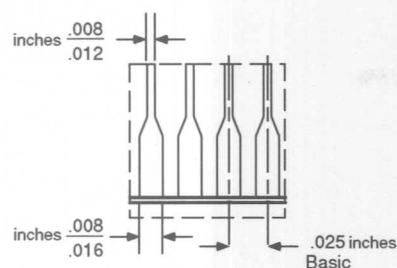
1. All dimensions are in inches unless otherwise specified.
2. Dimensions do not include mold protrusion.
3. Coplanarity of all leads will be within 0.004 inches measured from the seating plan. Coplanarity is measured per specification 06-500.
4. Lead spacing as measured from centerline to centerline will be within 0.003 inches.

**PQB132—Plastic Quad Flat Pack with Bumper, Molded Carrier Ring**  
(Inner device is measured in inches; outer ring is measured in millimeters)



16423A  
CB 50  
2/5/92 SG

## PQB 132—Plastic Quad Flat Pack with Bumper, Molded Carrier Ring (continued)



## Notes:

1. All dimensions and tolerances conform to ANSI Y14.5M-1982.
2. Controlling dimensions: package is measured in inches and ring is measured in millimeters.
3. D and E dimensions do not include mold protrusion. Allowable mold protrusion is 0.2 mm per side.
4. D, D3, E, and E3 dimensions include mold mismatch and are measured at the parting line.
5. Dimensions are centered about centerline of lead material.
6. Dimensions D1 and E1 are from outside edge to outside edge of the test points.
7. There are six locating holes in the ring. -B- and -C- datum holes are used for trim form and excise of the molded package only. Holes Z1 and Z2 are used for electrical testing only.
8. This area is reserved for vacuum pickup on each of the four corners of the ring and must be flat within .025 mm. No ejector pins in this area.
9. Surface A is used for seating in socket applications.
10. Pin one orientation with respect to carrier ring as indicated.



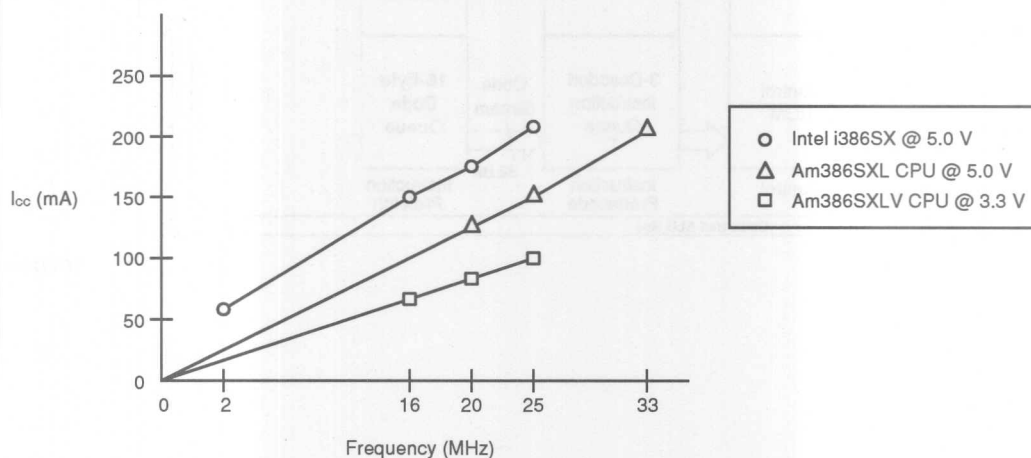


# Am386SXLV

High-Performance, Low-Voltage, 32-Bit  
Microprocessor with 16-Bit Data Bus

## DISTINCTIVE CHARACTERISTICS

- **Operating range 3.0 V to 5.5 V—Ideal for notebook PC designs**
  - 25-MHz operating frequency for 3.0 V–5.5 V
  - 33-MHz operating frequency for 4.5 V–5.5 V
  - Twice the battery life of existing 5-V designs
  - Wide range of chipsets and other logic available for 3-V systems with support for Standby Mode operation
  - True static design for long battery life
  - Power consumption 75% lower than Intel i386SX, 65% lower than Am386SXL microprocessor
  - Performance on demand (0 to 25 MHz)
- **System Management Mode (SMM) for system and power management**
  - System Management Interrupt (SMI) for power management independent of processor operating mode and operating system
  - SMI coupled with I/O instruction break feature provides transparent power off and auto resume of peripherals which may not be “power aware”
  - SMI is non-maskable and has higher priority than Non-Maskable Interrupt (NMI)
  - Automatic save and restore of the microprocessor state
  - Wide range of chipsets supporting SMM available to allow product differentiation
- **Lower heat dissipation for fanless systems**
- **“Float” input to facilitate system debug and test**
- **Compatible with 386SX systems and software**
- **Supports 387SX-compatible math coprocessors**
- **100-pin PQFP package with optional protective ring for better lead coplanarity**
- **AMD advanced 0.8 micron CMOS technology**

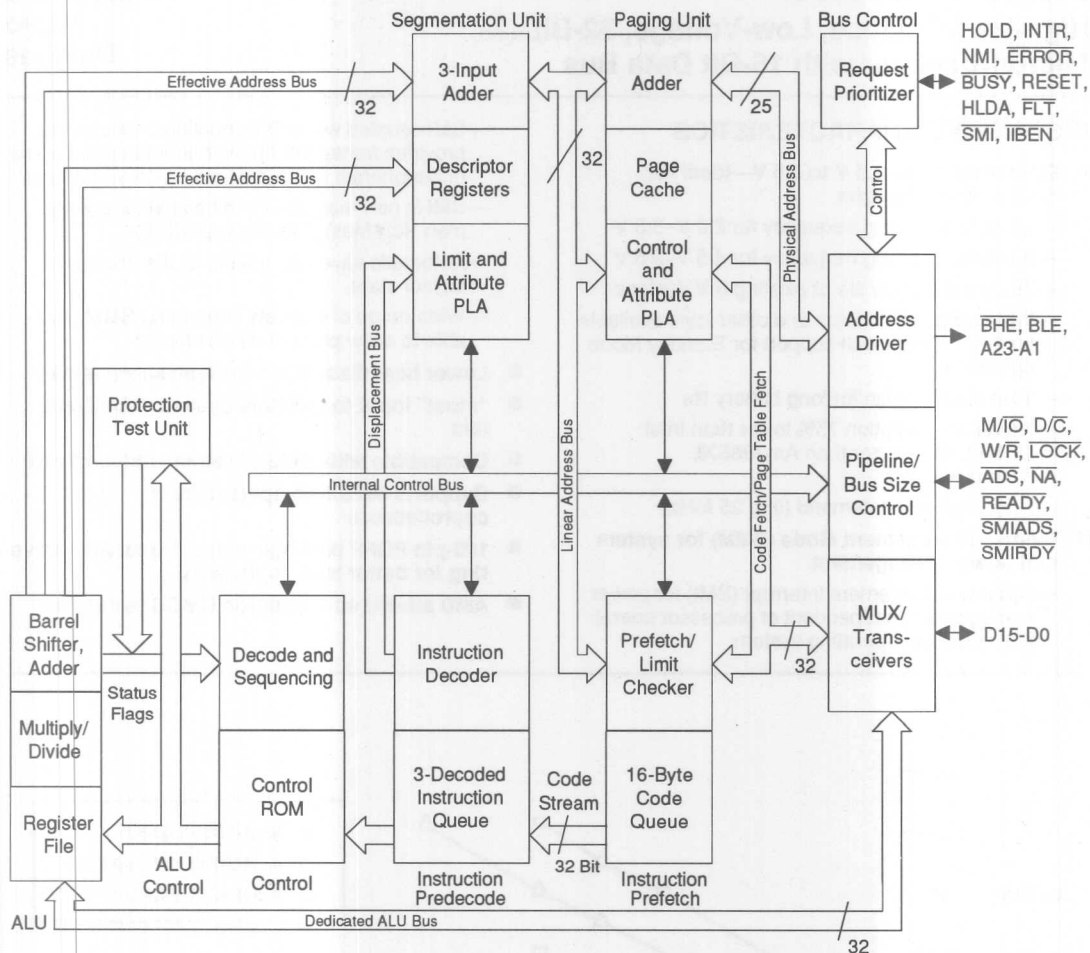


Note: Inputs at  $V_{CC}$  or  $V_{SS}$ .

## Typical Power Consumption

This document contains information on a product under development at Advanced Micro Devices, Inc. The information is intended to help you evaluate this product. AMD reserves the right to change or discontinue work on this proposed product without notice.

BLOCK DIAGRAM



16305C-001





## GENERAL DESCRIPTION

The Am386SXLV microprocessor is a low-voltage, true static implementation of the Intel i386SX microprocessor. With the operating range of 3.0 V to 5.5 V, the Am386SXLV CPU is ideal for both desktop and battery-powered notebook personal computers. For desktop PCs, this device offers lower heat dissipation, allowing system designers to remove or reduce the size and cost of the cooling fan. The Am386SXLV microprocessor operates at a maximum speed of 25 MHz from 3.0 V to 5.5 V and at maximum speed of 33 MHz from 4.5 V to 5.5 V.

The Am386SXLV microprocessor's lower operating voltage and true static design enables longer battery life

and/or lower weight for notebook applications. At 20 MHz, this device has 60% lower operating  $I_{CC}$  than the Intel i386SX. Lowering typical operating voltage from 5.0 V to 3.3 V doubles the battery life. Standby Mode allows the Am386SXLV microprocessor to be clocked down to 0 MHz (DC) and retain full register contents. In Standby Mode, typical current draw is less than 0.01 mA, a greater than 1000X reduction in power consumption versus the Intel i386SX.

The Am386SXLV microprocessor is available in a small footprint 100-pin Plastic Quad Flat Pack (PQFP) package. This package may be shipped in an optional protective ring for better lead protection during shipping.

Additionally, the Am386SXLV microprocessor comes with System Management Mode (SMM) for system and power management. SMI (System Management Interrupt) is a non-maskable, higher priority interrupt than NMI and has its own code space (1 Mb in Real Mode and 16 Mb in Protected Mode). SMI can be coupled with the I/O instruction break feature to implement transparent power management of peripherals. SMM can be used by system designers to implement system and power management code independent of the operating system or the processor mode.

The Am386SXLV microprocessor incorporates a float pin that places all outputs in a three-state mode to facilitate board test and debug.

## FUNCTIONAL DESCRIPTION

### Benefits of Lower Operating Voltage

The Am386SXLV microprocessor has an operating voltage range of 3.0 V to 5.5 V. Low voltage allows for lower operating power consumption, longer battery life, and/or smaller batteries for notebook applications.

Because power is proportional to the square of the voltage, reduction of the supply voltage from 5.0 V to 3.3 V reduces power consumption by 56%. This directly translates to a doubling of battery life for portable applications. Lower power consumption can also be used to reduce the size and weight of the battery. Thus, 3.3-V designs facilitate a reduction in the form factor.

A lower operating voltage results in a reduction of I/O voltage swings. This reduces noise generation and provide a less hostile environment for board design. Lower Operating Voltage also reduces electromagnetic radiation noise and makes to obtain FCC approval easier to obtain.

### SMM—System Management Mode

The Am386SXLV microprocessor has a new System Management Mode (SMM) for system and power management. This mode consists of two features: System Management Interrupt (SMI) and I/O instruction break.

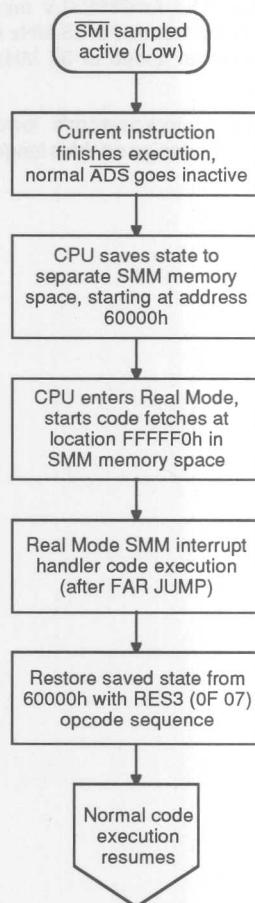
#### SMI—System Management Interrupt

SMI is implemented by using special bus interface pins. This interrupt method can be used to perform system management functions such as power management independent of processor operating mode (Real, Protected, or Virtual 8086 Modes).

SMI can also be invoked in software. This allows system software to communicate with SMI power management code. In addition, the UMOV instruction allows data transfers between SMI and normal system memory spaces.

Activating the  $\overline{\text{SMI}}$  pin invokes a sequence that saves the operating state of the processor into a separate SMM memory space, independent of the main system memory. After the state is saved, the processor is forced into Real Mode and begins execution at address

FFFFF0h in the SMM memory space where a far jump to the SMM code is executed. This Real Mode code can perform its system management function and then resume execution of the normal system software by executing an RES3 instruction which will reload the saved processor state and continue execution in the main system memory space. See Figure 1 for a general flow-chart of an SMM operation.



16305C-002

Figure 1. SMM Flow

### CPU Interface—Pin Functions

The CPU interface for SMM consists of three pins dedicated to the SMI function. One pin,  $\overline{\text{SMI}}$ , is the new interrupt input. The other two pins,  $\overline{\text{SMIADS}}$  and  $\overline{\text{SMIRDY}}$ , provide the control signals necessary for the separate SMM mode memory space.

### Description of SMM Operation

The execution of a System Management Interrupt has four distinct phases: the initiation of the interrupt via  $\overline{\text{SMI}}$ , a processor state save, execution of the SMM

interrupt code, and a processor state restore (to resume normal operation).

### Interrupt Initiation

A System Management Interrupt is initiated by the driving of a synchronous, active Low pulse on the  $\overline{\text{SMI}}$  pin until the first  $\overline{\text{SMIADS}}$  is asserted. This pulse period will ensure recognition of the interrupt. The CPU drives the  $\overline{\text{SMI}}$  pin active after the completion of the current operation (active bus cycle, instruction execution, or both). The active drive of the pin by the CPU is released at the end of the interrupt routine following the last register read of the saved state. The CPU drives  $\overline{\text{SMI}}$  High for two CLK2 cycles prior to releasing the drive of  $\overline{\text{SMI}}$ .

An SMI cannot be masked off by the CPU, and it will always be recognized by the CPU, regardless of operating modes. This includes the Real, Protected, and Virtual 8086 Modes of the processor.

While the CPU is in SMM, a bus hold request via the HOLD pin is granted. The HLDA pin goes active after bus release and the  $\overline{\text{SMIADS}}$  pin floats along with the other pins that normally float during a bus hold cycle. SMI does not float during a Bus Hold cycle.

### Processor State Save

The first set of SMM bus transfer cycles after the CPU's recognition of an active SMI is the processor saving its state to an external RAM array in a separate address space from main system memory. This is accomplished by using the  $\overline{\text{SMIADS}}$  and  $\overline{\text{SMIRDY}}$  pins for initiation and termination of bus cycles, instead of the  $\overline{\text{ADS}}$  and  $\overline{\text{READY}}$  pins. The 24-bit addresses to which the CPU saves its state are 60000h–600CBh and 60100h–60127h. These are fixed address locations for each register saved.

To ensure valid operation, pipelining must be disabled while the processor is in SMM. There are 114 data transfer cycles.

### SMI Code Execution

After the processor state is saved to the separate SMM memory space, the execution of the SMI interrupt routine code begins. The processor enters Real Mode, sets most of the register values to "reset" values (those values normally seen after a CPU reset), and begins fetching code from address FFFF0h in the separate SMM memory space. Normally, the first thing the interrupt routine code does is a FAR JUMP to the Real Mode entry point for the SMI interrupt routine, which is also in SMM memory space.

Both INTR and NMI are disabled upon entry into SMM. The SMM code can be located anywhere within the 1-Mb Real Mode address space, except for where the processor state is saved. I/O cycles, as a result of the IN, OUT, INS, and OUTS instructions, will go to the

normal address space, utilizing the normal  $\overline{\text{ADS}}$  and  $\overline{\text{READY}}$  bus interface signals. This facilitates power management code manipulating system hardware registers as needed through the standard I/O subsystem; a separate I/O space is not implemented.

### Processor State Restore (Resuming Normal Execution)

Returning to normal code execution in the main system memory, including restoring the processor operating mode, is accomplished by executing a special code sequence. This code invokes a restore CPU state operation that reloads the CPU registers from the saved data in the RAM controlled by  $\overline{\text{SMIADS}}$  and  $\overline{\text{SMIRDY}}$ .

The ES:EDI register pair must point to the physical address of the processor save state (6000h). In Real Mode the address is calculated as  $\text{ES} \cdot 16 + \text{EDI offset}$ . The saved state should not cross a 64K boundary. The RES3 instruction (0F 07) should be executed to start the restore state operation. After completion of the restore state operation, the  $\overline{\text{SMI}}$  pin will be deactivated by the CPU and normal code execution will continue at the point where it left off before the SMI occurred. There are 114 data transfer cycles in the restore operation.

### Software Features

Several features of the SMI function provide support for special operations during the execution of the system's software. These features involve the execution of reserved opcodes to induce specific SMI related operations.

### Software SMI Generation

Besides hardware initiation of the SMI via the  $\overline{\text{SMI}}$  pin, there is also a software induced SMI mechanism. Generating a soft SMI involves setting a control bit (Bit 12) in the Debug Control Register (DR7) and executing an SMI instruction (opcode F1h).

The functional sequence of the software-based SMI is identical to the hardware-based SMI with the exception that the  $\overline{\text{SMI}}$  pin is not initially driven active by an external source. Upon execution of a soft SMI opcode, the  $\overline{\text{SMI}}$  pin is driven active (Low) by the processor before the save state operation begins.

### Memory Transfers to Main System Memory

While executing an SMI routine, the interrupt code can initiate memory data reads and writes to the main system memory using the normal  $\overline{\text{ADS}}$  and  $\overline{\text{READY}}$  pins. This initiation is accomplished by using reserved opcodes that are special forms of the MOV instruction (called UMOV). The UMOV opcodes can move byte, word, or double word register operands to or from main system memory. Multiple data transfers using the normal  $\overline{\text{ADS}}$  and  $\overline{\text{READY}}$  pins will occur if the operands

are misaligned relative to the effective address used. The UMOV opcodes are 0F 10h, 0F 11h, 0F 12h, and 0F 13h. The UMOV instruction can use any of the 386 addressing modes, as specified in the ModR/M byte of the opcode. Note that the 16- and 32-bit versions are the same opcodes with the exception of the 66h operand size prefix.

## I/O Instruction Break

The Am386SXLV microprocessor has an I/O instruction break feature that allows the system logic to implement I/O trapping for peripheral devices. To enable the I/O Instruction break feature,  $\overline{\text{IIBEN}}$  must first be asserted active Low. On detecting an I/O instruction, the processor prevents the execution unit from executing further instructions until  $\overline{\text{READY}}$  is driven active Low by the system. Once  $\overline{\text{READY}}$  is driven active, the execution unit either immediately responds to any active interrupt request or continues executing instructions following the I/O instruction that caused the break.

The I/O instruction break feature can be used to allow system logic to implement I/O trapping for peripheral devices. On sensing an I/O instruction, the system drives the  $\overline{\text{SMI}}$  pin active before driving  $\overline{\text{READY}}$  active. This ensures that the interrupt service routine is executed immediately following the I/O instruction that caused the break. (If the I/O instruction break feature is not enabled via  $\overline{\text{IIBEN}}$ , several instructions could execute before the SMI service routine is executed.)

The SMI service routine can access the peripheral for which  $\overline{\text{SMI}}$  was asserted and modify its state. The SMI service routine normally returns to the instruction following the I/O instruction that caused the break. By modifying the saved state instruction pointer, the routine can choose to return to the I/O instruction that caused the break and re-execute that instruction. The default is to return to the following instruction (except for REP I/O string instruction). To re-execute the I/O instruction that caused the break, the SMI service routine must copy the I/O instruction pointer over the default pointer. This feature is particularly useful when an application program requests an access to a peripheral that has been powered down. The SMI service routine can restore power to the peripheral and initiate a re-execution sequence transparent to the application program. This re-execution feature should only be used if the SMI is in response to an I/O trap with  $\overline{\text{IIBEN}}$  active. Note that the I/O instruction break feature is not enabled for memory mapped I/O devices or for coprocessor bus cycles even if  $\overline{\text{IIBEN}}$  is active.

## I/O Instruction Break Timing

The I/O Instruction Break feature requires that  $\overline{\text{SMI}}$  be sampled active (Low) by the processor at least three CLK2 edges before the CLK2 edge that ends the I/O cycle with an active  $\overline{\text{READY}}$  signal. This timing applies for

both pipelined and non-pipelined cycles. If this timing constraint is not met, additional instructions may be executed by the internal execution unit prior to entering SMM. Depending on the state of the prefetch queue at the time the SMI is asserted, instruction fetch cycles may occur on the normal  $\overline{\text{ADS}}$  interface before the SMM save state process begins with the assertion of  $\overline{\text{SMIADS}}$ . However, this fetched code will not be executed.

## True Static Operation

The Am386SXLV microprocessor incorporates a true static design. Unlike dynamic circuit design, the Am386SXLV device eliminates the minimum operating frequency restriction. It may be clocked from its maximum speed all the way down to 0 MHz (DC). System designers can use this feature to design battery-powered notebook PCs with long battery life.

## Standby Mode

This true static design of the Am386SXLV microprocessor allows for a Standby Mode. At any operating speed the Am386SXLV microprocessor will retain its state (i.e., the contents of all of its registers). By shutting off the clock completely, the device enters Standby Mode. Since power consumption is proportional to clock frequency, operating power consumption is reduced as the frequency is lowered. In Standby Mode, typical current draw is reduced to less than 0.01 mA. Not only does this feature save battery life, but it also simplifies the design of power-conscious notebook computers in the following ways.

1. Eliminates the need for software in BIOS to save and restore the contents of registers.
2. Allows simpler circuitry to control stopping of the clock (since) the system does not need to know the state of the processor.

## Lower Operating Icc

True static design also allows lower operating  $I_{cc}$  when operating at any speed.

## Performance on Demand

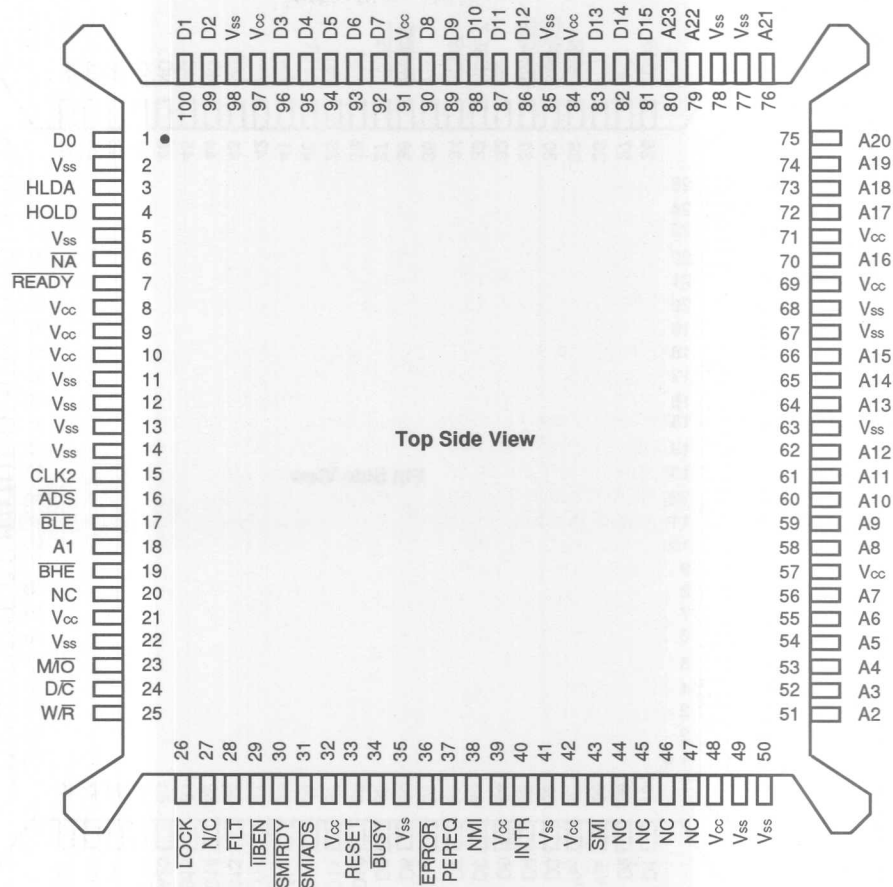
The Am386SXLV microprocessor retains its state at any speed from 0 MHz (DC) to its maximum operating speed. With this feature, system designers may vary the operating speed of the system to extend the battery life in notebook systems.

For example, the system could operate at low speeds during inactivity or polling operations. However, upon interrupt, the system clock can be increased up to its maximum speed. After a user-defined time-out period, the system can be returned to a low (or 0 MHz) operating speed without losing its state. This design maximizes battery life while achieving optimal performance.



## CONNECTION DIAGRAM

## 100-Lead Plastic Quad Flat Pack (PQFP) Package—Top Side View

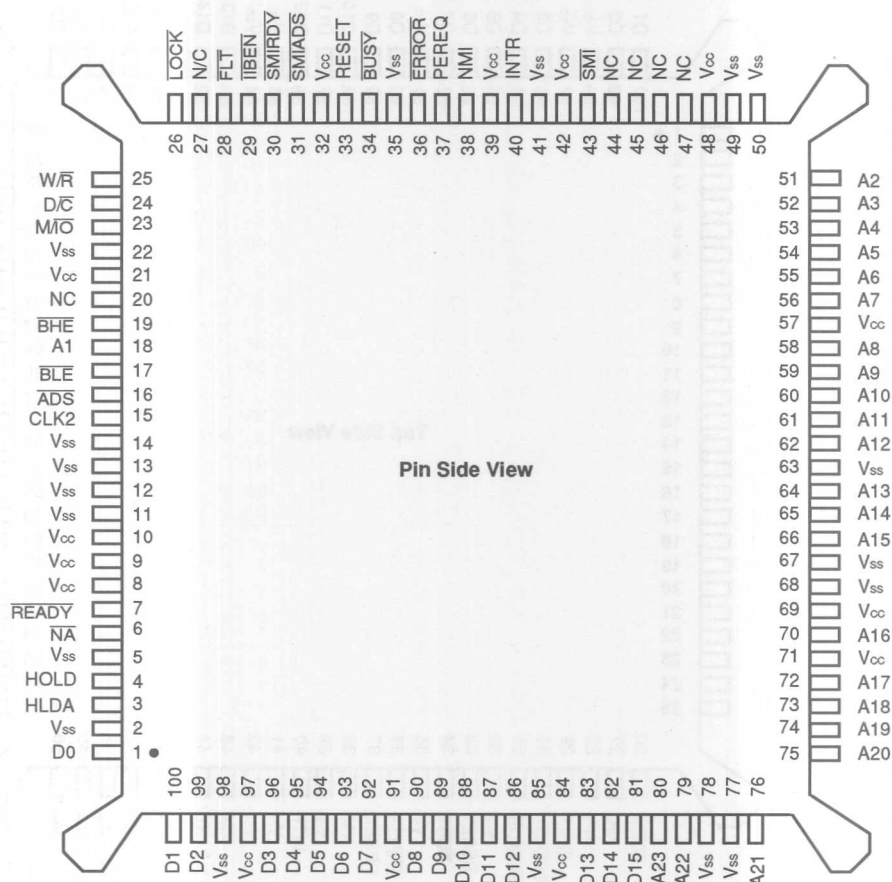


Notes: Pin 1 is marked for orientation.

N/C = Not connected; connection of an N/C pin may cause a malfunction or incompatibility with future shippings of the Am386SXLV microprocessor.

# CONNECTION DIAGRAM

## 100-Lead Plastic Quad Flat Pack (PQFP) Package—Pin Side View



Notes: Pin 1 is marked for orientation.

N/C = Not connected; connection of an N/C pin may cause a malfunction or incompatibility with future shippings of the Am386SXLV microprocessor.



## PIN DESIGNATION TABLES (Sorted by Functional Grouping)

| Address  |         | Data     |         | Control  |         | NC      | V <sub>CC</sub> | V <sub>SS</sub> |
|----------|---------|----------|---------|----------|---------|---------|-----------------|-----------------|
| Pin Name | Pin No. | Pin Name | Pin No. | Pin Name | Pin No. | Pin No. | Pin No.         | Pin No.         |
| A1       | 18      | D0       | 1       | ADS      | 16      | 20      | 8               | 2               |
| A2       | 51      | D1       | 100     | BHE      | 19      | 27      | 9               | 5               |
| A3       | 52      | D2       | 99      | BLE      | 17      | 44      | 10              | 11              |
| A4       | 53      | D3       | 96      | BUSY     | 34      | 45      | 21              | 12              |
| A5       | 54      | D4       | 95      | CLK2     | 15      | 46      | 32              | 13              |
| A6       | 55      | D5       | 94      | D/C      | 24      | 47      | 39              | 14              |
| A7       | 56      | D6       | 93      | ERROR    | 36      |         | 42              | 22              |
| A8       | 58      | D7       | 92      | FLT      | 28      |         | 48              | 35              |
| A9       | 59      | D8       | 90      | HLDA     | 3       |         | 57              | 41              |
| A10      | 60      | D9       | 89      | HOLD     | 4       |         | 69              | 49              |
| A11      | 61      | D10      | 88      | IIBEN    | 29      |         | 71              | 50              |
| A12      | 62      | D11      | 87      | INTR     | 40      |         | 84              | 63              |
| A13      | 64      | D12      | 86      | LOCK     | 26      |         | 91              | 67              |
| A14      | 65      | D13      | 83      | M/IO     | 23      |         | 97              | 68              |
| A15      | 66      | D14      | 82      | NA       | 6       |         |                 | 77              |
| A16      | 70      | D15      | 81      | NMI      | 38      |         |                 | 78              |
| A17      | 72      |          |         | PEREQ    | 37      |         |                 | 85              |
| A18      | 73      |          |         | READY    | 7       |         |                 | 98              |
| A20      | 75      |          |         | RESET    | 33      |         |                 |                 |
| A21      | 76      |          |         | SMI      | 43      |         |                 |                 |
| A22      | 79      |          |         | SMIADS   | 31      |         |                 |                 |
| A23      | 80      |          |         | SMIRDY   | 30      |         |                 |                 |
|          |         |          |         | W/R      | 25      |         |                 |                 |

## PIN DESIGNATION TABLES (Sorted by Pin Number)

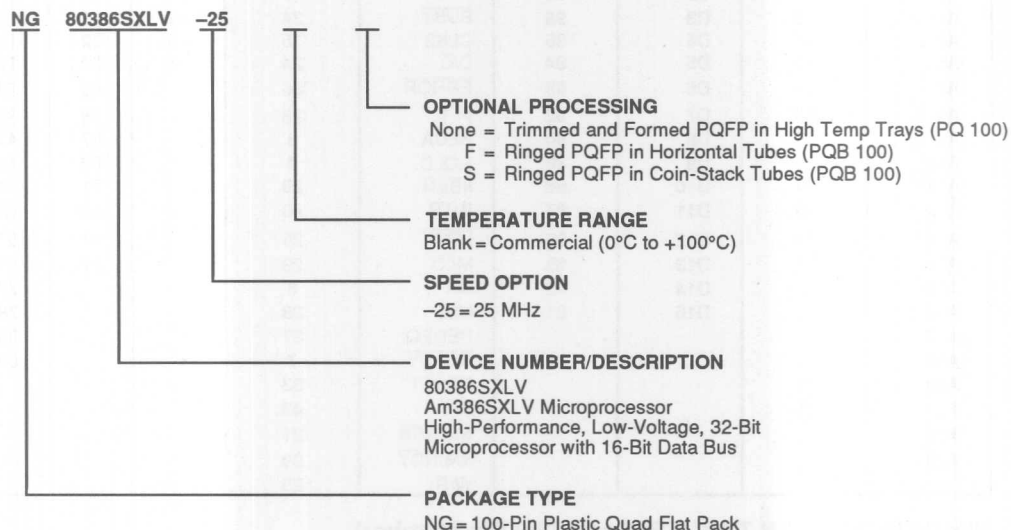
| Pin No. | Pin Name        | Pin No. | Pin Name        | Pin No. | Pin Name        | Pin No. | Pin Name        | Pin No. | Pin Name        |
|---------|-----------------|---------|-----------------|---------|-----------------|---------|-----------------|---------|-----------------|
| 1       | D0              | 21      | V <sub>CC</sub> | 41      | V <sub>SS</sub> | 61      | A11             | 81      | D15             |
| 2       | V <sub>SS</sub> | 22      | V <sub>SS</sub> | 42      | V <sub>CC</sub> | 62      | A12             | 82      | D14             |
| 3       | HLDA            | 23      | M/IO            | 43      | SMI             | 63      | V <sub>SS</sub> | 83      | D13             |
| 4       | HOLD            | 24      | D/C             | 44      | NC              | 64      | A13             | 84      | V <sub>CC</sub> |
| 5       | V <sub>SS</sub> | 25      | W/R             | 45      | NC              | 65      | A14             | 85      | V <sub>SS</sub> |
| 6       | NA              | 26      | LOCK            | 46      | NC              | 66      | A15             | 86      | D12             |
| 7       | READY           | 27      | NC              | 47      | NC              | 67      | V <sub>SS</sub> | 87      | D11             |
| 8       | V <sub>CC</sub> | 28      | FLT             | 48      | V <sub>CC</sub> | 68      | V <sub>SS</sub> | 88      | D10             |
| 9       | V <sub>CC</sub> | 29      | IIBEN           | 49      | V <sub>SS</sub> | 69      | V <sub>CC</sub> | 89      | D9              |
| 10      | V <sub>CC</sub> | 30      | SMIRDY          | 50      | V <sub>SS</sub> | 70      | A16             | 90      | D8              |
| 11      | V <sub>SS</sub> | 31      | SMIADS          | 51      | A2              | 71      | V <sub>CC</sub> | 91      | V <sub>CC</sub> |
| 12      | V <sub>SS</sub> | 32      | V <sub>CC</sub> | 52      | A3              | 72      | A17             | 92      | D7              |
| 13      | V <sub>SS</sub> | 33      | RESET           | 53      | A4              | 73      | A18             | 93      | D6              |
| 14      | V <sub>SS</sub> | 34      | BUSY            | 54      | A5              | 74      | A19             | 94      | D5              |
| 15      | CLK2            | 35      | V <sub>SS</sub> | 55      | A6              | 75      | A20             | 95      | D4              |
| 16      | ADS             | 36      | ERROR           | 56      | A7              | 76      | A21             | 96      | D3              |
| 17      | BLE             | 37      | PEREQ           | 57      | V <sub>CC</sub> | 77      | V <sub>SS</sub> | 97      | V <sub>CC</sub> |
| 18      | A1              | 38      | NMI             | 58      | A8              | 78      | V <sub>SS</sub> | 98      | V <sub>SS</sub> |
| 19      | BHE             | 39      | V <sub>CC</sub> | 59      | A9              | 79      | A22             | 99      | D2              |
| 20      | NC              | 40      | INTR            | 60      | A10             | 80      | A23             | 100     | D1              |



## ORDERING INFORMATION

### Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of the elements below.

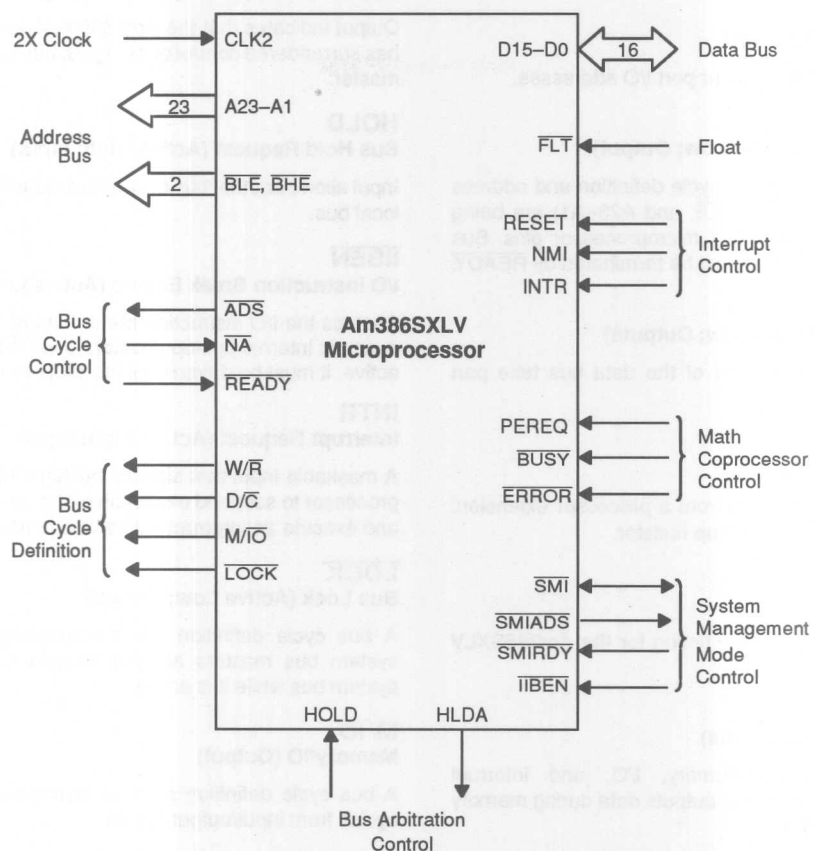


| Valid Combinations |           |      |
|--------------------|-----------|------|
| NG                 | 80386SXLV | -25  |
|                    |           | -25F |
|                    |           | -25S |

#### Valid Combinations

Valid Combinations lists configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

## LOGIC SYMBOL



16305C-003

## PIN DESCRIPTIONS

### A23-A1

#### Address Bus (Outputs)

Outputs physical memory or port I/O addresses.

### ADS

#### Address Status (Active Low; Output)

Indicates that a valid bus cycle definition and address ( $\overline{W/R}$ ,  $\overline{D/C}$ ,  $\overline{M/I/O}$ ,  $\overline{BHE}$ ,  $\overline{BLE}$ , and A23-A1) are being driven at the Am386SXLV microprocessor pins. Bus cycles initiated by ADS must be terminated by READY.

### $\overline{BHE}$ , $\overline{BLE}$

#### Byte Enables (Active Low; Outputs)

Indicate which data bytes of the data bus take part in a bus cycle.

### BUSY

#### Busy (Active Low; Input)

Signals a busy condition from a processor extension. BUSY has an internal pull-up resistor.

### CLK2

#### CLK2 (Input)

Provides the fundamental timing for the Am386SXLV microprocessor.

### D15-D0

#### Data Bus (Inputs/Outputs)

Inputs data during memory, I/O, and interrupt acknowledge read cycles; outputs data during memory and I/O write cycles.

### $\overline{D/C}$

#### Data/Control (Output)

A bus cycle definition pin that distinguishes data cycles, either memory or I/O, from control cycles which are: interrupt acknowledge, halt, and code fetch.

### ERROR

#### Error (Active Low; Input)

Signals an error condition from a processor extension. ERROR has an internal pull-up resistor.

### FLT

#### Float (Active Low; Input)

An input which forces all bidirectional and output signals, including HLDA, to the three-state condition. FLT has an internal pull-up resistor. The pin, if not used, should be disconnected.

### HLDA

#### Bus Hold Acknowledge (Active High; Output)

Output indicates that the Am386SXLV microprocessor has surrendered control of its logical bus to another bus master.

### HOLD

#### Bus Hold Request (Active High; Input)

Input allows another bus master to request control of the local bus.

### $\overline{IIBEN}$

#### I/O Instruction Break Enable (Active Low; Input)

Enables the I/O instruction break feature.  $\overline{IIBEN}$  has a dynamic internal pull-up resistor. Once  $\overline{IIBEN}$  is driven active, it must be driven until the processor is reset.

### INTR

#### Interrupt Request (Active High; Input)

A maskable input that signals the Am386SXLV microprocessor to suspend execution of the current program and execute an interrupt acknowledge function.

### LOCK

#### Bus Lock (Active Low; Output)

A bus cycle definition pin that indicates that other system bus masters are not to gain control of the system bus while it is active.

### $\overline{M/I/O}$

#### Memory/IO (Output)

A bus cycle definition pin that distinguishes memory cycles from input/output cycles.

### $\overline{NA}$

#### Next Address (Active Low; Input)

Used to request address pipelining.

### N/C

#### No Connect

Should always be left unconnected. Connection of an N/C pin may cause the processor to malfunction or be incompatible with future steppings of the Am386SXLV microprocessor.

### NMI

#### Non-Maskable Interrupt Request (Active High; Input)

A non-maskable input that signals the Am386SXLV microprocessor to suspend execution of the current program and execute an interrupt acknowledge function.

**PEREQ****Processor Extension Request (Active High; Input)**

Indicates that the processor has data to be transferred by the Am386SXLV microprocessor. PEREQ has an internal pull-down resistor.

**READY****Bus Ready (Active Low; Input)**

Terminates the bus cycle initiated by  $\overline{\text{ADS}}$ .

**RESET****Reset (Active High; Input)**

Suspends any operation in progress and places the Am386SXLV microprocessor in a known reset state.

**SMI****System Management Interrupt (Active Low; I/O)**

A non-maskable interrupt pin that signals the Am386SXLV microprocessor to suspend execution and enter System Management Mode. SMI has an internal pull-up resistor. SMI has a dynamic internal pull-up resistor that is disabled when the processor is in SMM. SMI is not three-stated during Hold Acknowledge bus cycles.

**SMIADS****SMI Address Status (Active Low; Output)**

When active, this pin indicates that a valid bus cycle definition and address (W/ $\overline{\text{R}}$  D/C, M/I/O, BHE, BLE, and

A23–A1) are being driven at the Am386SXLV microprocessor pins while in the System Management Mode. Bus cycles initiated by SMIADS must be terminated by SMIRDY.

**SMIRDY****SMI Ready (Active Low; Input)**

This input terminates the current bus cycle to the SMM Mode address space in the same manner the  $\overline{\text{READY}}$  pin does for the normal mode address space. SMIRDY has an internal pull-up resistor.  $\overline{\text{READY}}$  and SMIRDY must not be tied together.

**V<sub>cc</sub>****System Power (Input)**

Provides the DC supply input.

**V<sub>ss</sub>****System Ground (Input)**

Provides the 0-V connection from which all inputs and outputs are measured.

**W/ $\overline{\text{R}}$** **Write/Read (Output)**

A bus cycle definition pin that distinguishes write cycles from read cycles.

# ABSOLUTE MAXIMUM RATINGS

Storage Temperature . . . . .  $-65^{\circ}\text{C}$  to  $+150^{\circ}\text{C}$   
Ambient Temperature Under Bias . .  $-65^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$

*Stresses above those listed may cause permanent damage to the device. Functionality at or above these limits is not implied. Exposure to ABSOLUTE MAXIMUM RATING conditions for extended periods of time may affect device reliability.*

# OPERATING RANGES

Supply Voltage with respect to  $V_{SS}$  . .  $-0.5\text{ V}$  to  $+7.0\text{ V}$   
Voltage on Other Pins . . . . .  $-0.5\text{ V}$  to  $V_{CC} + 0.5\text{ V}$

*Operating ranges define those limits between which the functionality of the device is guaranteed.*

# DC CHARACTERISTICS over COMMERCIAL operating ranges

$V_{CC} = 3.0\text{ V}$  to  $3.6\text{ V}$ ;  $T_{CASE} = 0^{\circ}\text{C}$  to  $+100^{\circ}\text{C}$

| Symbol     | Parameter Description   | Notes   | Preliminary     |                                      | Unit                           |
|------------|---|---|-----------------|--------------------------------------|--------------------------------|
|            |   |   | Min             | Max                                  |                                |
| $V_{IL}$   | Input Low Voltage   | (Note 1)  | $-0.3$          | $+0.8$                               | V                              |
| $V_{IH}$   | Input High Voltage  |   | 2.0             | $V_{CC} + 0.3$                       | V                              |
| $V_{ILC}$  | CLK2 Input Low Voltage  | (Note 1)  | $-0.3$          | $+0.8$                               | V                              |
| $V_{IHC}$  | CLK2 Input High Voltage (25 MHz)  |   | 2.4             | $V_{CC} + 0.3$                       | V                              |
| $V_{OL}$   | Output Low Voltage  | (Note 5)  |                 |                                      |                                |
|            | $I_{OL} = 0.5\text{ mA}$ : A23–A1, D15–D0   |   |                 | 0.2                                  | V                              |
|            | $I_{OL} = 0.5\text{ mA}$ : BHE, BLE, W/R, D/C, SMIADS, M/I $\bar{O}$ , LOCK, ADS, HLDA, SMI       |   |                 | 0.2                                  | V                              |
|            | $I_{OL} = 2\text{ mA}$ : A23–A1, D15–D0   |   |                 | 0.45                                 | V                              |
| $V_{OH}$   | Output High Voltage   | (Note 5)<br>(Note 6)  |                 |                                      |                                |
|            | $I_{OH} = 0.1\text{ mA}$ : A23–A1, D15–D0   |   | $V_{CC} - 0.2$  |                                      | V                              |
|            | $I_{OH} = 0.1\text{ mA}$ : BHE, BLE, W/R, D/C, SMIADS, LOCK, ADS, M/I $\bar{O}$ , HLDA, SMI       |   | $V_{CC} - 0.2$  |                                      | V                              |
|            | $I_{OH} = 0.5\text{ mA}$ : A23–A1, D15–D0   |   | $V_{CC} - 0.45$ |                                      | V                              |
| $I_{LI}$   | Input Leakage Current (All pins except PEREQ, BUSY, ERROR, SMI, SMIRDY, FLT, IIBEN)               | $0\text{ V} \leq V_{IN} \leq V_{CC}$<br>(Note 7)                                |                 | $\pm 10$                             | $\mu\text{A}$                  |
|            |   |   |                 |                                      |                                |
| $I_{IH}$   | Input Leakage Current (PEREQ pin)   | $V_{IH} = V_{CC} - 0.1\text{ V}$<br>$V_{IH} = 2.4\text{ V}$ (Note 2)            |                 | 300<br>200                           | $\mu\text{A}$<br>$\mu\text{A}$ |
| $I_{IL}$   | Input Leakage Current (BUSY, ERROR, SMI, SMIRDY, FLT, IIBEN)                                      | $V_{IL} = 0.1\text{ V}$<br>$V_{IL} = 0.45\text{ V}$ (Note 3)                    |                 | $-300$<br>$-200$                     | $\mu\text{A}$<br>$\mu\text{A}$ |
| $I_{LO}$   | Output Leakage Current  | $0.1\text{ V} \leq V_{OUT} \leq V_{CC}$   |                 | $\pm 15$                             | $\mu\text{A}$                  |
| $I_{CC}$   | Supply Current (Note 8)<br>CLK2 = 40 MHz: Oper. Freq. 20 MHz<br>CLK2 = 50 MHz: Oper. Freq. 25 MHz | $V_{CC} = 3.3\text{ V}$<br>$I_{CC}\text{ Typ} = 80$<br>$I_{CC}\text{ Typ} = 95$ |                 | $V_{CC} = 3.6\text{ V}$<br>95<br>115 | $\text{mA}$<br>$\text{mA}$     |
| $I_{CCSB}$ | Standby Current (Note 8)  | $I_{CCSB}\text{ Typ} = 10\text{ }\mu\text{A}$                                   |                 | 150                                  | $\mu\text{A}$                  |
| $C_{IN}$   | Input or I/O Capacitance  | $F_C = 1\text{ MHz}$ (Note 4)   |                 | 10                                   | pF                             |
| $C_{OUT}$  | Output Capacitance  | $F_C = 1\text{ MHz}$ (Note 4)   |                 | 12                                   | pF                             |
| $C_{CLK}$  | CLK2 Capacitance  | $F_C = 1\text{ MHz}$ (Note 4)   |                 | 20                                   | pF                             |

- Notes: 1. The Min value,  $-0.3$ , is not 100% tested.  
2. PEREQ input has an internal pull-down resistor.  
3. BUSY, ERROR, FLT, SMI, IIBEN, and SMIRDY inputs each have an internal pull-up resistor.  
4. Not 100% tested.  
5. Outputs are CMOS and will pull rail-to-rail if the load is not resistive.  
6.  $V_{OH}$  SMI only valid on SMI output when exiting SMM for two CLK2 periods.  
7. SMI and IIBEN leakage Low will be  $I_{LI}$  when pull-up is inactive and  $I_{IL}$  when pull-up is active.  
8. Inputs at rails ( $V_{CC}$  or  $V_{SS}$ ).

**ABSOLUTE MAXIMUM RATINGS**

Storage Temperature . . . . . -65°C to +150°C

Ambient Temperature under Bias . . -65°C to +125°C

Stresses above those listed may cause permanent damage to the device. Functionality at or above these limits is not implied. Exposure to ABSOLUTE MAXIMUM RATING conditions for extended periods of time may affect device reliability.

**OPERATING RANGES**Supply Voltage with respect to  $V_{SS}$  . . . -0.5 V to +7 VVoltage on Other Pins . . . . . -0.5 V to  $V_{CC} + 0.5$  V

Operating ranges define those limits between which the functionality of the device is guaranteed.

**DC CHARACTERISTICS over COMMERCIAL operating ranges** $V_{CC} = 4.5$  V to 5.5 V;  $T_{CASE} = 0^\circ$  C to +100°C

| Symbol     | Parameter Description  | Notes  | Preliminary    |                | Unit    |
|------------|--|--|----------------|----------------|---------|
|            |  |  | Min            | Max            |         |
| $V_{IL}$   | Input Low Voltage  | (Note 1)   | -0.3           | +0.8           | V       |
| $V_{IH}$   | Input High Voltage   |  | 2.0            | $V_{CC} + 0.3$ | V       |
| $V_{ILC}$  | CLK2 Input Low Voltage   | (Note 1)   | -0.3           | +0.8           | V       |
| $V_{IHC}$  | CLK2 Input High Voltage (33 MHz)   |  | 2.7            | $V_{CC} + 0.3$ | V       |
| $V_{OL}$   | Output Low Voltage<br>$I_{OL} = 4$ mA: A23-A1, D15-D0<br>$I_{OL} = 5$ mA: BHE, BLE, W/R, D/C, SMIADS, M/I/O, LOCK, ADS, HLDA, SMI  | (Note 5)   |                | 0.45           | V       |
|            |  |  |                | 0.45           | V       |
| $V_{OH}$   | Output High Voltage<br>$I_{OH} = 1.0$ mA: A23-A1, D15-D0<br>$I_{OH} = 0.2$ mA: A23-A1, D15-D0<br>$I_{OH} = 0.9$ mA: BHE, BLE, W/R, D/C, SMIADS, LOCK, ADS, M/I/O, HLDA, SMI<br>$I_{OH} = 0.18$ mA: BHE, BLE, W/R, D/C, SMIADS, LOCK, ADS, M/I/O, HLDA, SMI | (Note 5)   | 2.4            |                | V       |
|            |  |  | $V_{CC} - 0.5$ |                | V       |
|            |  | (Note 6)   | 2.4            |                | V       |
|            |  |  | $V_{CC} - 0.5$ |                | V       |
| $I_L$      | Input Leakage Current (All pins except PEREQ, BUSY, ERROR, SMI, SMIRDY, FLT, and IIBEN)  | $0 \text{ V} \leq V_{IN} \leq V_{CC}$<br>(Note 7)                                      |                | $\pm 15$       | $\mu$ A |
| $I_{IH}$   | Input Leakage Current (PEREQ pin)  | $V_{IH} = 2.4$ V (Note 2)  |                | 200            | $\mu$ A |
| $I_{IL}$   | Input Leakage Current (BUSY, ERROR, SMI, SMIRDY, FLT, IIBEN)   | $V_{IL} = 0.45$ V (Note 3)   |                | -400           | $\mu$ A |
| $I_{LO}$   | Output Leakage Current   | $0.1 \text{ V} \leq V_{OUT} \leq V_{CC}$   |                | $\pm 15$       | $\mu$ A |
| $I_{CC}$   | Supply Current (Note 8)<br>CLK2 = 40 MHz: Oper. Freq. 20 MHz<br>CLK2 = 50 MHz: Oper. Freq. 25 MHz<br>CLK2 = 66 MHz: Oper. Freq. 33 MHz   | $V_{CC}$ Typ = 5.0 V<br>$I_{CC}$ Typ = 130<br>$I_{CC}$ Typ = 160<br>$I_{CC}$ Typ = 210 |                | $V_{CC} = 5.5$ | V       |
|            |  |  |                | 155            | mA      |
|            |  |  |                | 190            | mA      |
|            |  |  |                | 245            | mA      |
| $I_{CCSB}$ | Standby Current (Note 8)   | $I_{CCSB}$ Typ = 20 $\mu$ A  |                | 150            | $\mu$ A |
| $C_{IN}$   | Input or I/O Capacitance   | $F_C = 1$ MHz (Note 4)   |                | 10             | pF      |
| $C_{OUT}$  | Output Capacitance   | $F_C = 1$ MHz (Note 4)   |                | 12             | pF      |
| $C_{CLK}$  | CLK2 Capacitance   | $F_C = 1$ MHz (Note 4)   |                | 20             | pF      |

Notes: 1. The Min value, -0.3, is not 100% tested.

2. PEREQ input has an internal pull-down resistor.

3. BUSY, ERROR, SMI, IIBEN, and SMIRDY inputs each have an internal pull-up resistor.

4. Not 100% tested.

5. Outputs are CMOS and will pull rail-to-rail if the load is not resistive.

6.  $V_{OH}$  SMI only valid on SMI output when exiting SMM for two CLK2 periods.7. SMI and IIBEN leakage Low will be  $I_L$  when pull-up is inactive and  $I_{IL}$  when pull-up is active.8. Inputs at rails ( $V_{CC}$  or  $V_{SS}$ ).

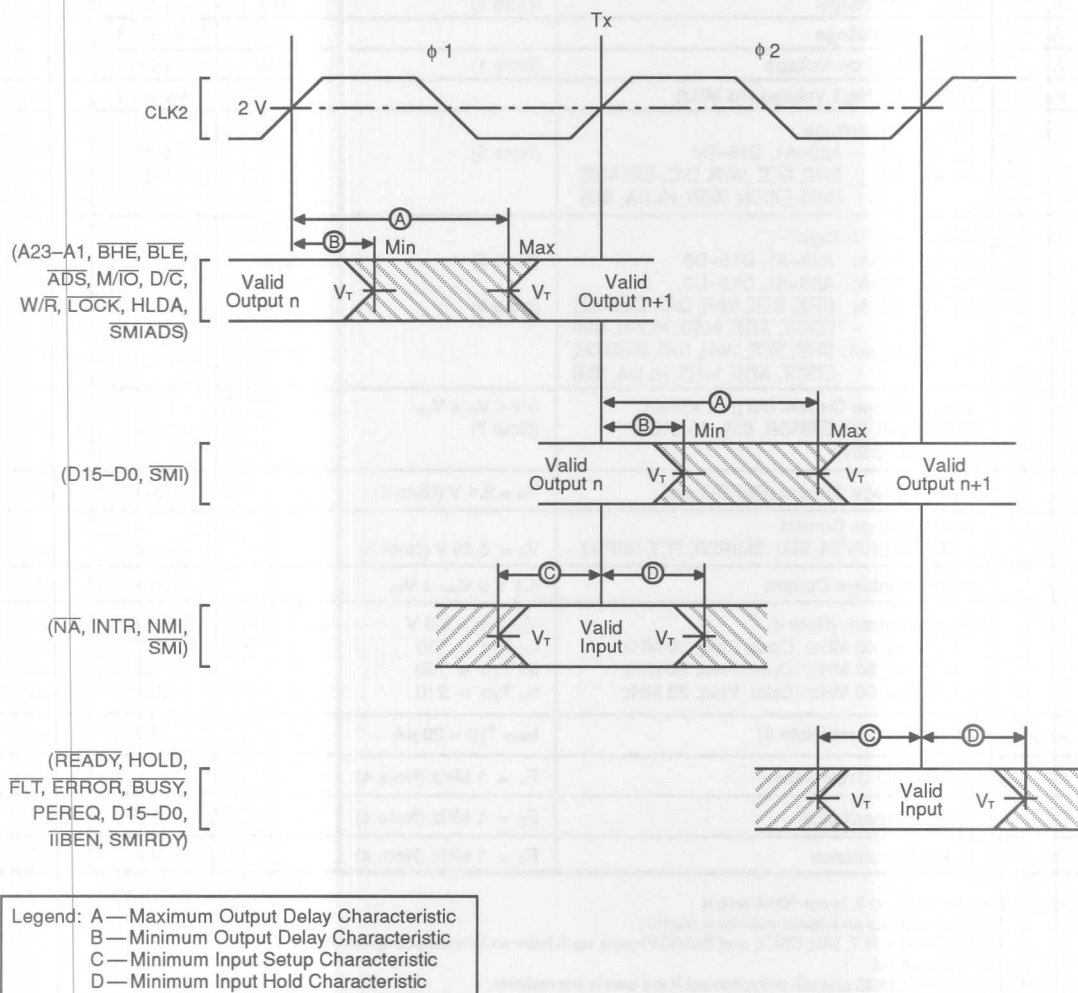
## SWITCHING CHARACTERISTICS

The switching characteristics given consist of output delays, input setup requirements, and input hold requirements. All switching characteristics are relative to the CLK2 rising edge crossing the 2.0-V level.

Switching characteristic measurement is defined in Figure 2. Inputs must be driven to the voltage levels indicated by Figure 2 when switching characteristics are measured. Output delays are specified with minimum and maximum limits measured, as shown. The minimum delay times are hold times provided to external circuitry. Input setup and hold times are specified as

minimums, defining the smallest acceptable sampling window. Within the sampling window, a synchronous input signal must be stable for correct operation.

Outputs  $\overline{ADS}$ ,  $W/\overline{R}$ ,  $D/\overline{C}$ ,  $M/\overline{IO}$ ,  $LOCK$ ,  $BHE$ ,  $\overline{BLE}$ ,  $\overline{SMIADS}$ ,  $A23-A1$ , and  $HLDA$  only change at the beginning of phase one.  $D15-D0$  and  $\overline{SMI}$  write cycles only change at the beginning of phase two. The  $\overline{IBEN}$ ,  $READY$ ,  $HOLD$ ,  $BUSY$ ,  $\overline{SMIRDY}$ ,  $ERROR$ ,  $PEREQ$ ,  $FLT$ , and  $D15-D0$  (read cycles) inputs are sampled at the beginning of phase one. The  $\overline{NA}$ ,  $INTR$ ,  $NMI$ , and  $\overline{SMI}$  inputs are sampled at the beginning of phase two.



Notes: 1. Input waveforms have  $t_r \leq 2.0$  ns from 0.8 V to 2.0 V.  
2.  $V_t = 1.0$  V for  $V_{cc} \leq 3.6$  V; 1.5 V for  $V_{cc} > 3.6$  V.

16305C-003

Figure 2. Drive Levels and Measurement Points for Switching Characteristics



**SWITCHING CHARACTERISTICS over operating ranges at 25 MHz** $V_{CC} = 3.0 \text{ V to } 3.6 \text{ V}$ ;  $T_{CASE} = 0^{\circ}\text{C to } 100^{\circ}\text{C}$ 

| Symbol | Parameter Description   | Notes                   | Ref. Figures | Preliminary |     | Unit |
|--------|---|-------------------------|--------------|-------------|-----|------|
|        |   |                         |              | Min         | Max |      |
|        | Operating Frequency   | Half CLK2 freq.         |              | 0           | 25  | MHz  |
| 1      | CLK2 Period   |                         | 3            | 20          |     | ns   |
| 2      | CLK2 High Time  | at $V_{IH}$             | 3            | 4           |     | ns   |
| 3      | CLK2 Low Time   | at 0.8 V                | 3            | 5           |     | ns   |
| 4      | CLK2 Fall Time  | 2.4 V to 0.8 V (Note 3) | 3            |             | 7   | ns   |
| 5      | CLK2 Rise Time  | 0.8 V to 2.4 V (Note 3) | 3            |             | 7   | ns   |
| 6      | A23–A1 Valid Delay  | $C_L = 50 \text{ pF}$   | 6            | 4           | 17  | ns   |
| 7      | A23–A1 Float Delay  | (Note 1)                | 13           | 4           | 30  | ns   |
| 8      | BHE, BLE, LOCK Valid Delay  | $C_L = 50 \text{ pF}$   | 6            | 4           | 17  | ns   |
| 9      | BHE, BLE, LOCK Float Delay  | (Note 1)                | 13           | 4           | 30  | ns   |
| 10     | M/ $\overline{IO}$ , D/ $\overline{C}$ , W/ $\overline{R}$ , $\overline{ADS}$ Valid Delay | $C_L = 50 \text{ pF}$   | 6            | 4           | 17  | ns   |
| 10s    | SMI $\overline{ADS}$ Valid Delay  | $C_L = 50 \text{ pF}$   | 6            | 4           | 25  | ns   |
| 11     | W/ $\overline{R}$ , M/ $\overline{IO}$ , D/ $\overline{C}$ , $\overline{ADS}$ Float Delay | (Note 1)                | 13           | 4           | 30  | ns   |
| 11s    | SMI $\overline{ADS}$ Float Delay  | (Note 1)                | 13           | 4           | 30  | ns   |
| 12     | D15–D0 Write Data Valid Delay   | $C_L = 50 \text{ pF}$   | 6, 7         | 7           | 23  | ns   |
| 12a    | D15–D0 Write Data Hold Time   | $C_L = 50 \text{ pF}$   | 8            | 2           |     | ns   |
| 13     | D15–D0 Write Data Float Delay   | (Note 1)                | 13           | 4           | 22  | ns   |
| 14     | HLDA Valid Delay  | $C_L = 50 \text{ pF}$   | 6            | 4           | 22  | ns   |
| 14f    | HLDA Float Delay  | (Notes 1, 4)            | 14           | 4           | 30  | ns   |
| 15     | $\overline{NA}$ Setup Time  |                         | 5            | 5           |     | ns   |
| 16     | $\overline{NA}$ Hold Time   |                         | 5            | 3           |     | ns   |
| 19     | READY Setup Time  |                         | 5            | 9           |     | ns   |
| 19s    | SMIRDY Setup Time   |                         | 5            | 9           |     | ns   |
| 20     | READY Hold Time   |                         | 5            | 4           |     | ns   |
| 20s    | SMIRDY Hold Time  |                         | 5            | 4           |     | ns   |
| 21     | D15–D0 Read Data Setup Time   |                         | 5            | 7           |     | ns   |
| 22     | D15–D0 Read Data Hold Time  |                         | 5            | 5           |     | ns   |
| 23     | HOLD Setup Time   |                         | 5            | 9           |     | ns   |
| 24     | HOLD Hold Time  |                         | 5            | 3           |     | ns   |
| 25     | RESET Setup Time  |                         | 15           | 8           |     | ns   |
| 26     | RESET Hold Time   |                         | 15           | 3           |     | ns   |
| 27     | NMI, INTR Setup Time  | (Note 2)                | 5            | 6           |     | ns   |
| 27s    | SMI Setup Time  |                         | 5            | 6           |     | ns   |
| 28     | NMI, INTR Hold Time   | (Note 2)                | 5            | 6           |     | ns   |
| 28s    | SMI Hold Time   |                         | 5            | 4           |     | ns   |
| 29     | PEREQ, ERROR, BUSY, FLT, IIBEN Setup Time   | (Note 2)                | 5            | 6           |     | ns   |
| 30     | PEREQ, ERROR, BUSY, FLT, IIBEN Hold Time  | (Note 2)                | 5            | 5           |     | ns   |
| 31     | SMI Valid Delay   |                         | 6, 13        | 4           | 22  | ns   |
| 32     | SMI Float Delay   | (Notes 1, 4)            | 14           | 4           | 30  | ns   |

- Notes: 1. Float condition occurs when maximum output current becomes less than  $I_{LO}$  in magnitude. Float delay is not 100% tested.  
 2. These inputs are allowed to be asynchronous to CLK2. The setup and hold specifications are given for testing purposes, to assure recognition within a specific CLK2 period.  
 3. Rise and Fall are not tested. They are guaranteed by design characterization.  
 4. Only during FLT assertion.



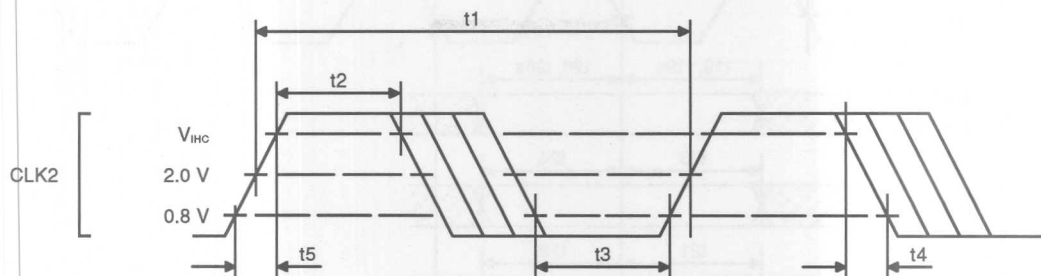
## SWITCHING CHARACTERISTICS over operating ranges at 33 MHz

 $V_{CC} = 4.5\text{ V} - 5.5\text{ V}$ ;  $T_{CASE} = 0^{\circ}\text{C}$  to  $100^{\circ}\text{C}$ 

| Symbol | Parameter Description                     | Notes                   | Ref. Figures | Preliminary |      | Unit |
|--------|---|-------------------------|--------------|-------------|------|------|
|        |   |                         |              | Min         | Max  |      |
|        | Operating Frequency                       | Half CLK2 freq.         |              | 0           | 33.3 | MHz  |
| 1      | CLK2 Period                               |                         | 3            | 15          |      | ns   |
| 2      | CLK2 High Time                            | at $V_{IH}$             | 3            | 4.5         |      | ns   |
| 3      | CLK2 Low Time                             | at 0.8 V                | 3            | 4.5         |      | ns   |
| 4      | CLK2 Fall Time                            | 2.4 V to 0.8 V (Note 3) | 3            |             | 4    | ns   |
| 5      | CLK2 Rise Time                            | 0.8 V to 2.4 V (Note 3) | 3            |             | 4    | ns   |
| 6      | A23–A1 Valid Delay                        | $C_L = 50\text{ pF}$    | 6            | 4           | 15   | ns   |
| 7      | A23–A1 Float Delay                        | (Note 1)                | 13           | 4           | 20   | ns   |
| 8      | BHE, BLE, LOCK Valid Delay                | $C_L = 50\text{ pF}$    | 6            | 4           | 15   | ns   |
| 9      | BHE, BLE, LOCK Float Delay                | (Note 1)                | 13           | 4           | 20   | ns   |
| 10     | M/IO, D/C, W/R, ADS Valid Delay           | $C_L = 50\text{ pF}$    | 6            | 4           | 15   | ns   |
| 10s    | SMAADS Valid Delay                        | $C_L = 50\text{ pF}$    | 6            | 4           | 15   | ns   |
| 11     | W/R, M/IO, D/C, ADS Float Delay           | (Note 1)                | 13           | 4           | 20   | ns   |
| 11s    | SMAADS Float Delay                        | (Note 1)                | 13           | 4           | 20   | ns   |
| 12     | D15–D0 Write Data Valid Delay             | $C_L = 50\text{ pF}$    | 6, 7         | 7           | 23   | ns   |
| 12a    | D15–D0 Write Data Hold Time               | $C_L = 50\text{ pF}$    | 8            | 2           |      | ns   |
| 13     | D15–D0 Write Data Float Delay             | (Note 1)                | 13           | 4           | 17   | ns   |
| 14     | HLDA Valid Delay                          | $C_L = 50\text{ pF}$    | 6            | 4           | 20   | ns   |
| 14f    | HLDA Float Delay                          | (Notes 1, 4)            | 14           | 4           | 20   | ns   |
| 15     | NA Setup Time                             |                         | 5            | 5           |      | ns   |
| 16     | NA Hold Time                              |                         | 5            | 2           |      | ns   |
| 19     | READY Setup Time                          |                         | 5            | 7           |      | ns   |
| 19s    | SMIRDY Setup Time                         |                         | 5            | 7           |      | ns   |
| 20     | READY Hold Time                           |                         | 5            | 4           |      | ns   |
| 20s    | SMIRDY Hold Time                          |                         | 5            | 4           |      | ns   |
| 21     | D15–D0 Read Data Setup Time               |                         | 5            | 5           |      | ns   |
| 22     | D15–D0 Read Data Hold Time                |                         | 5            | 3           |      | ns   |
| 23     | HOLD Setup Time                           |                         | 5            | 9           |      | ns   |
| 24     | HOLD Hold Time                            |                         | 5            | 2           |      | ns   |
| 25     | RESET Setup Time                          |                         | 15           | 5           |      | ns   |
| 26     | RESET Hold Time                           |                         | 15           | 2           |      | ns   |
| 27     | NMI, INTR Setup Time                      | (Note 2)                | 5            | 5           |      | ns   |
| 27s    | SMI Setup Time                            |                         | 5            | 5           |      | ns   |
| 28     | NMI, INTR Hold Time                       | (Note 2)                | 5            | 5           |      | ns   |
| 28s    | SMI Hold Time                             |                         | 5            | 4           |      | ns   |
| 29     | PEREQ, ERROR, BUSY, FLT, IIBEN Setup Time | (Note 2)                | 5            | 5           |      | ns   |
| 30     | PEREQ, ERROR, BUSY, FLT, IIBEN Hold Time  | (Note 2)                | 5            | 4           |      | ns   |
| 31     | SMI Valid Delay                           |                         | 6, 13        | 4           | 17   | ns   |
| 32     | SMI Float Delay                           | (Notes 1, 4)            | 14           | 4           | 20   | ns   |

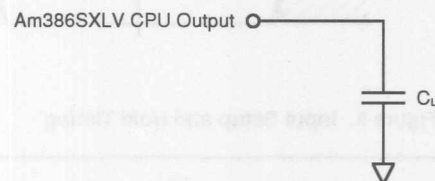
- Notes:
1. Float condition occurs when maximum output current becomes less than  $I_{LO}$  in magnitude. Float delay is not 100% tested.
  2. These inputs are allowed to be asynchronous to CLK2. The setup and hold specifications are given for testing purposes, to assure recognition within a specific CLK2 period.
  3. Rise and Fall are not tested. They are guaranteed by design characterization.
  4. Only during FLT assertion.

## SWITCHING CHARACTERISTICS (continued)



16305C-004

Figure 3. CLK2 Timing

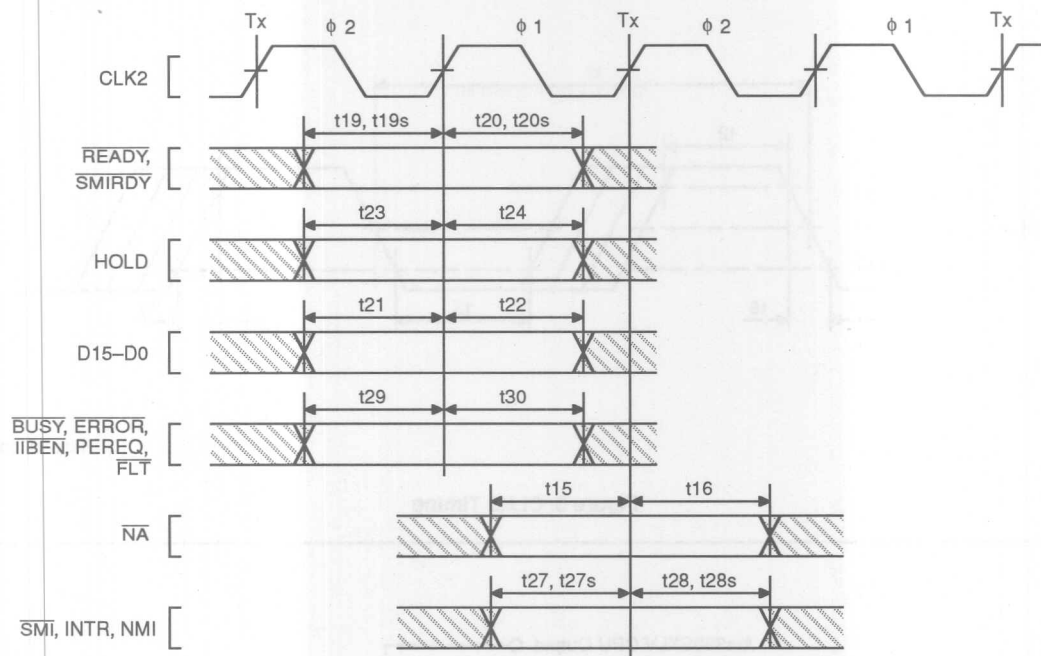


15022B-032

Figure 4. AC Test Circuit

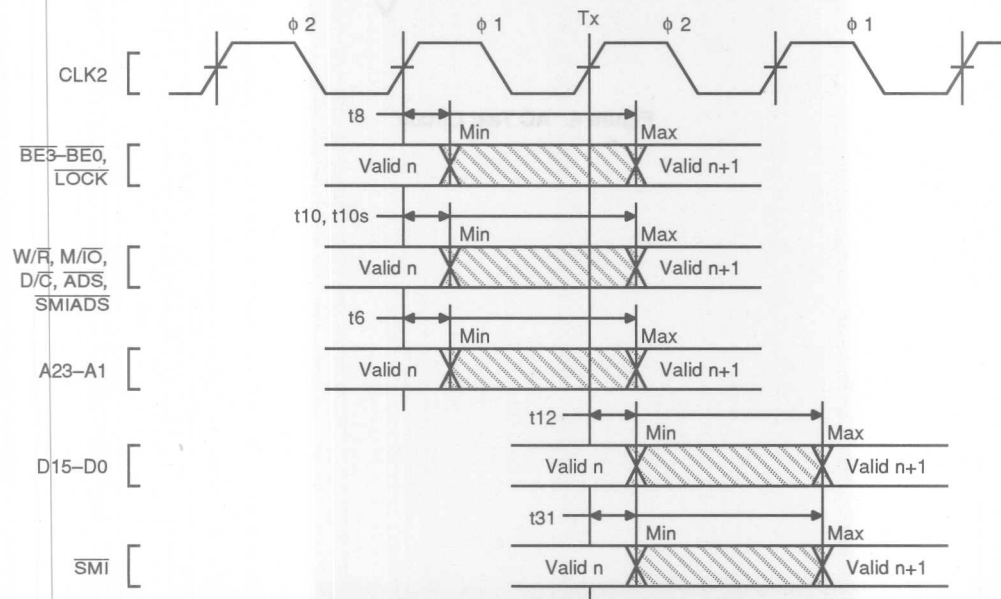


## SWITCHING WAVEFORMS



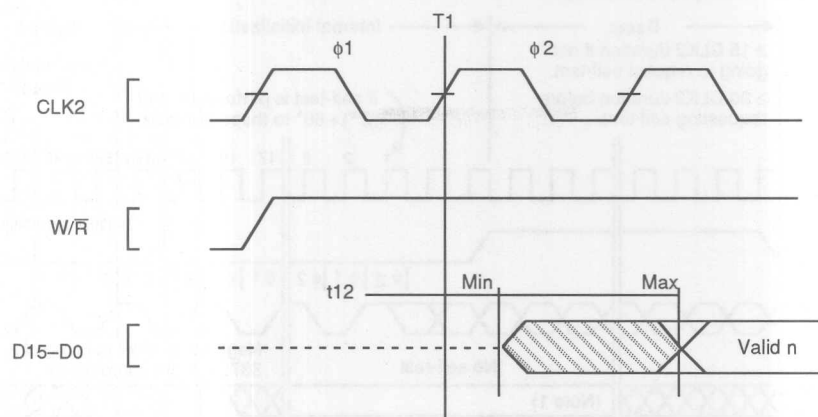
16305C-005

Figure 5. Input Setup and Hold Timing



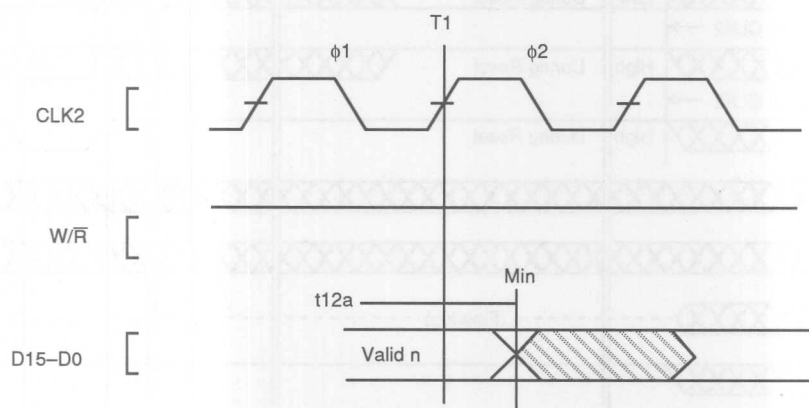
16305C-006

Figure 6. Output Valid Delay Timing



13605C-007

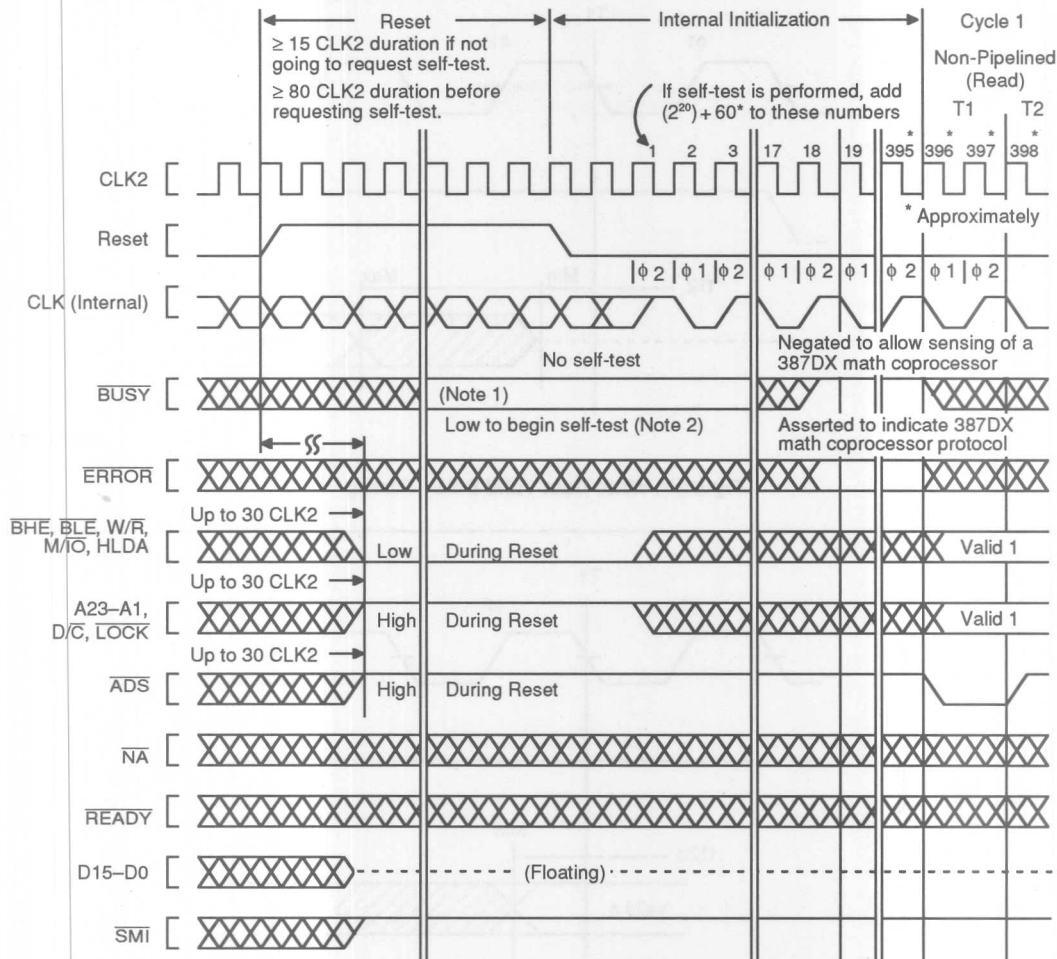
Figure 7. Write Data Valid Delay Timing



16305C-008

Figure 8. Write Data Hold Timing

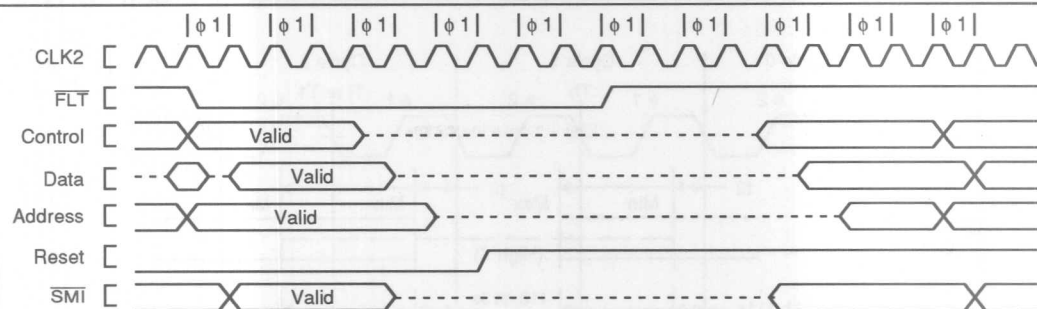
# SWITCHING WAVEFORMS (continued)



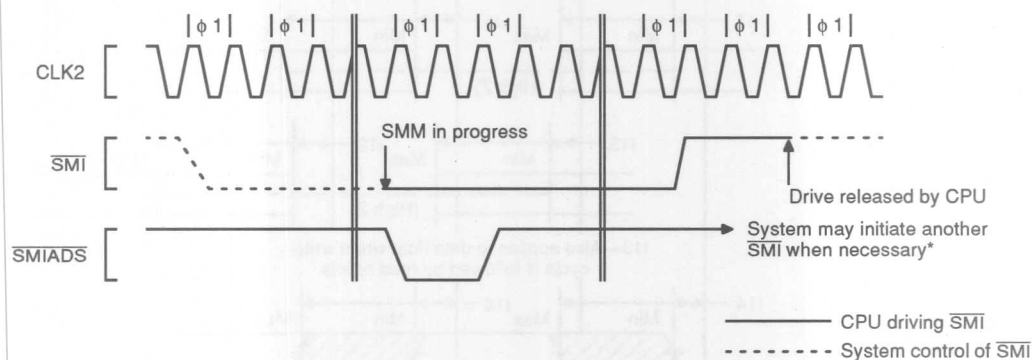
Notes: 1. BUSY should be held stable for eight CLK2 periods before and after the CLK2 period in which the RESET falling edge occurs.  
2. If self-test is requested, the Am386SXLV microprocessor outputs remain in their reset state as shown here.

Figure 9. Bus Activity from Reset Until First Code Fetch

16305C-009

Figure 10. Entering and Exiting  $\overline{\text{FLT}}$ 

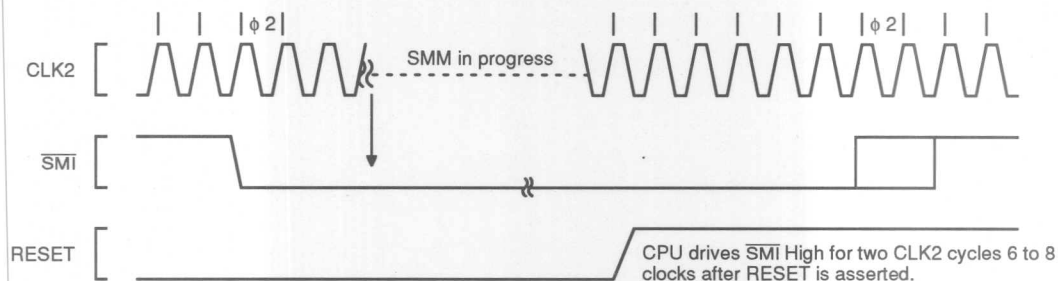
16306B-008



\*Once initiated, the system must hold  $\overline{\text{SMI}}$  Low until the first  $\overline{\text{SMIADS}}$ . At this time, the system cannot drive  $\overline{\text{SMI}}$  until three CLK2 cycles after the CPU drives  $\overline{\text{SMI}}$  High. (The CPU will drive  $\overline{\text{SMI}}$  High for two CLK2 cycles. The additional clock allows the CPU to completely release  $\overline{\text{SMI}}$  and prevents any driver overlap.)

16306B-011

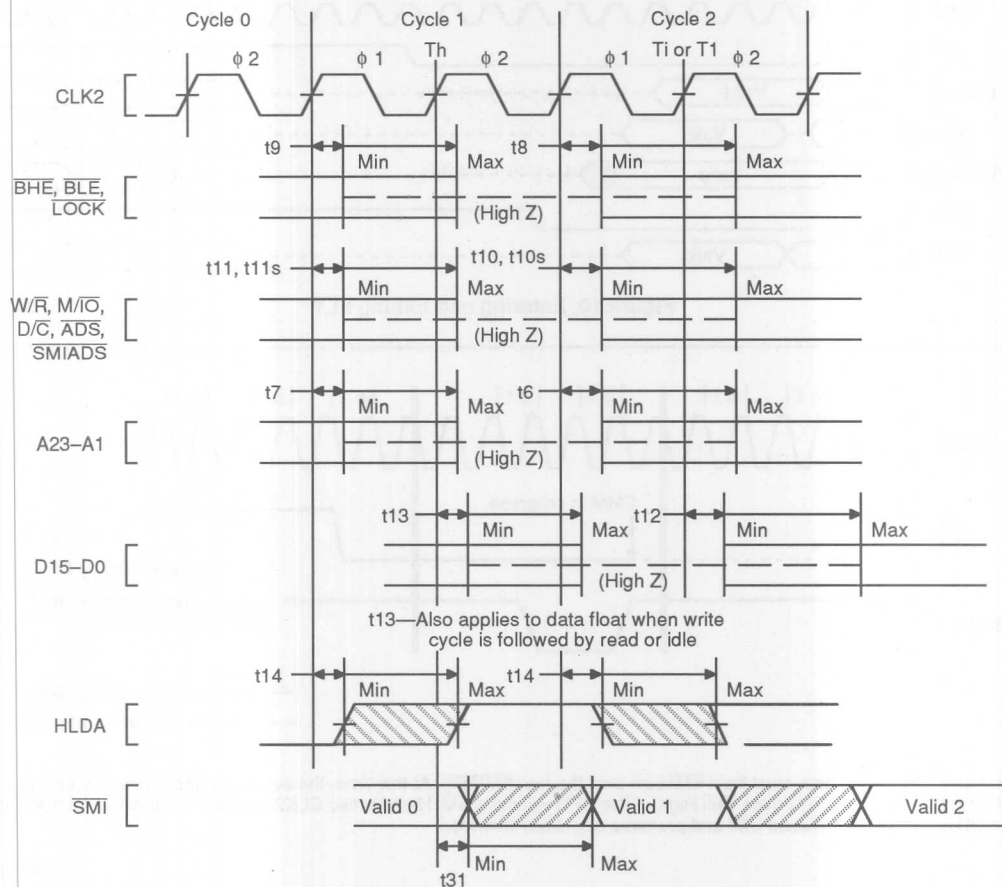
Figure 11. Initiating and Exiting SMM



16306B-010

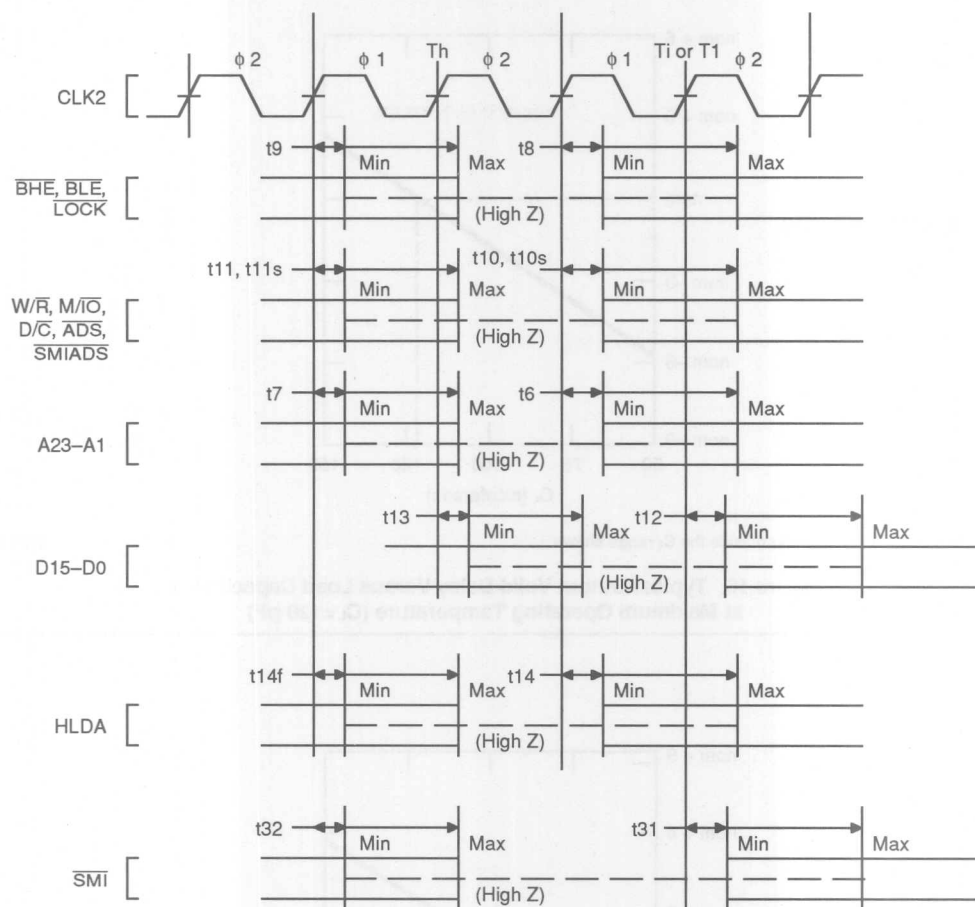
Figure 12. RESET and  $\overline{\text{SMI}}$



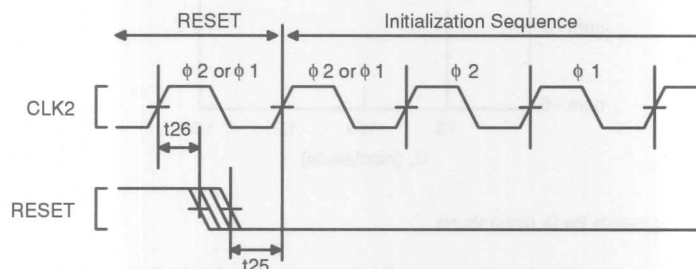


16305C-011

Figure 13. Output Float Delay and HLDA and  $\overline{\text{SMI}}$  Valid Delay Timing

Figure 14. Output Float Delay Entering and Exiting  $\overline{FLT}$ 

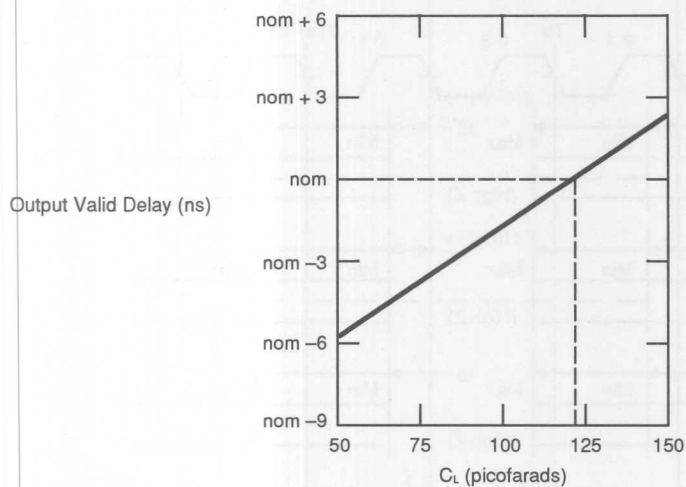
16305C-012



The second internal processor phase following RESET High-to-Low transition (provided  $t_{25}$  and  $t_{26}$  are met) is  $\phi 2$ .

15021B-084

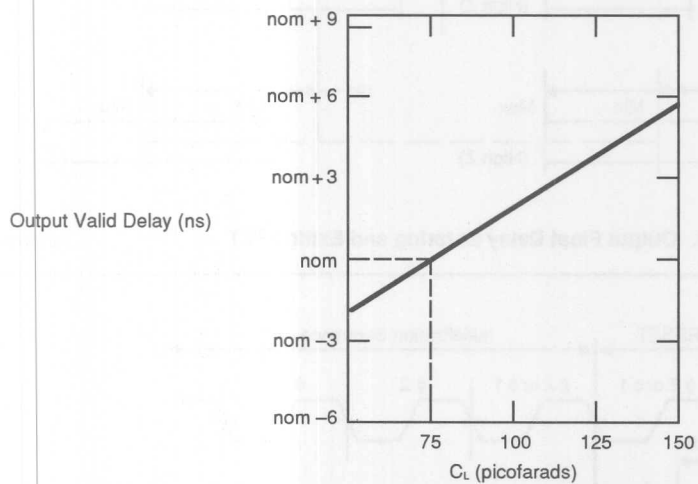
Figure 15. RESET Setup and Hold Timing and Internal Phase



Note: This graph will not be linear outside the  $C_L$  range shown.

15021B-079

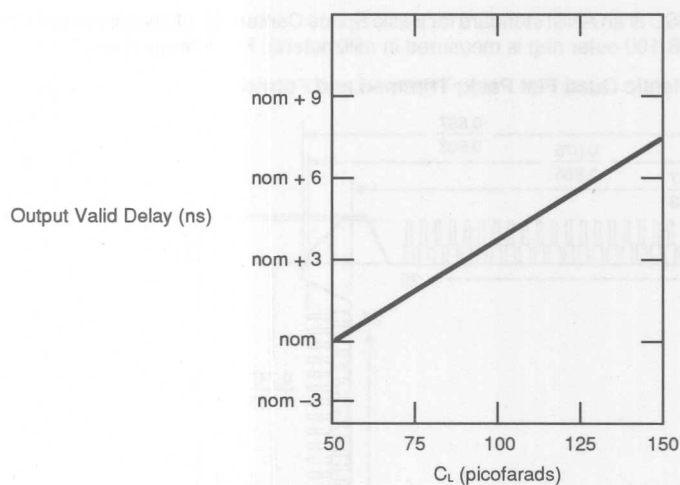
**Figure 16. Typical Output Valid Delay Versus Load Capacitance at Maximum Operating Temperature ( $C_L = 120$  pF)**



Note: This graph will not be linear outside the  $C_L$  range shown.

15021B-080

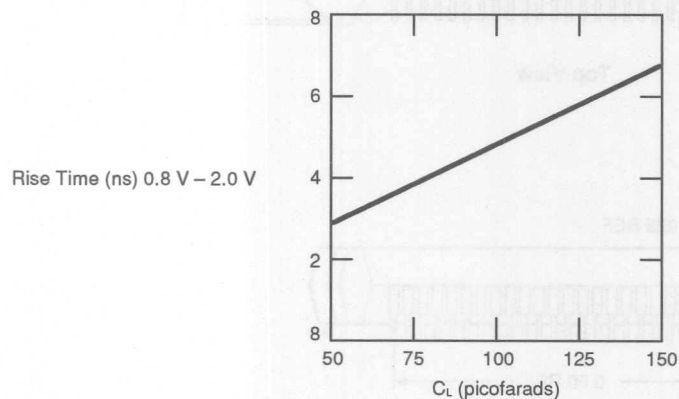
**Figure 17. Typical Output Valid Delay Versus Load Capacitance at Maximum Operating Temperature ( $C_L = 75$  pF)**



Note: This graph will not be linear outside the  $C_L$  range shown.

**Figure 18. Typical Output Valid Delay Versus Load Capacitance at Maximum Operating Temperature ( $C_L = 50$  pF)**

15021B-081



Note: This graph will not be linear outside the  $C_L$  range shown.

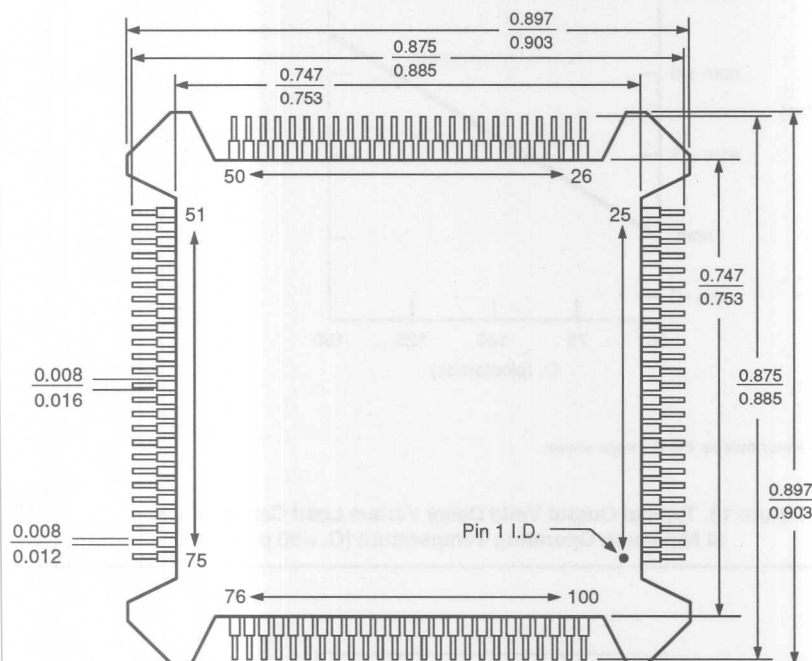
15021B-082

**Figure 19. Typical Output Rise Time Versus Load Capacitance at Maximum Operating Temperature**

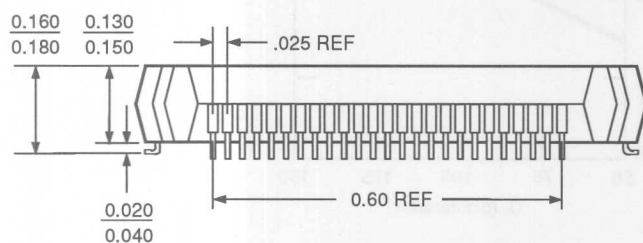
## PHYSICAL DIMENSIONS

Preliminary; package in development. BSC is an ANSI standard for Basic Space Centering. All measurements are in inches unless otherwise specified (PQB 100 outer ring is measured in millimeters). For reference only.

**PQ 100—Plastic Quad Flat Pack; Trimmed and Formed**



**Top View**

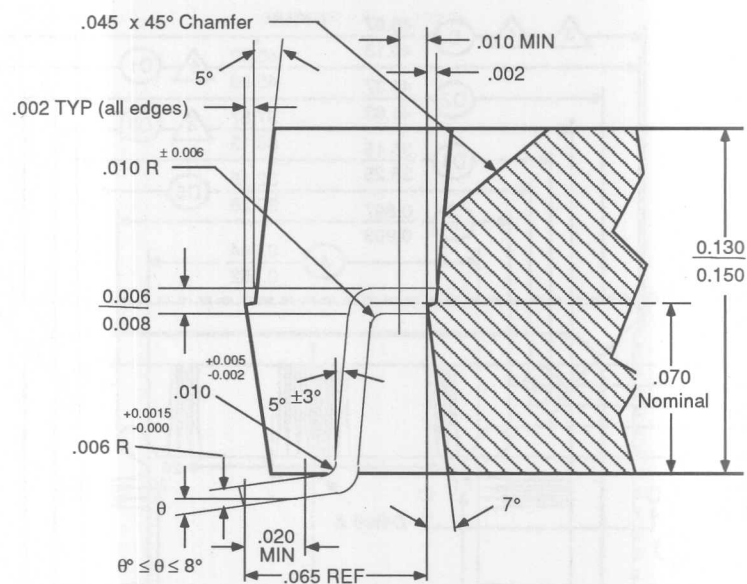


**Side View**

15679E  
BX 44  
3/31/92 SG

## PHYSICAL DIMENSIONS (continued)

## PQ 100—Plastic Quad Flat Pack; Trimmed and Formed (continued)



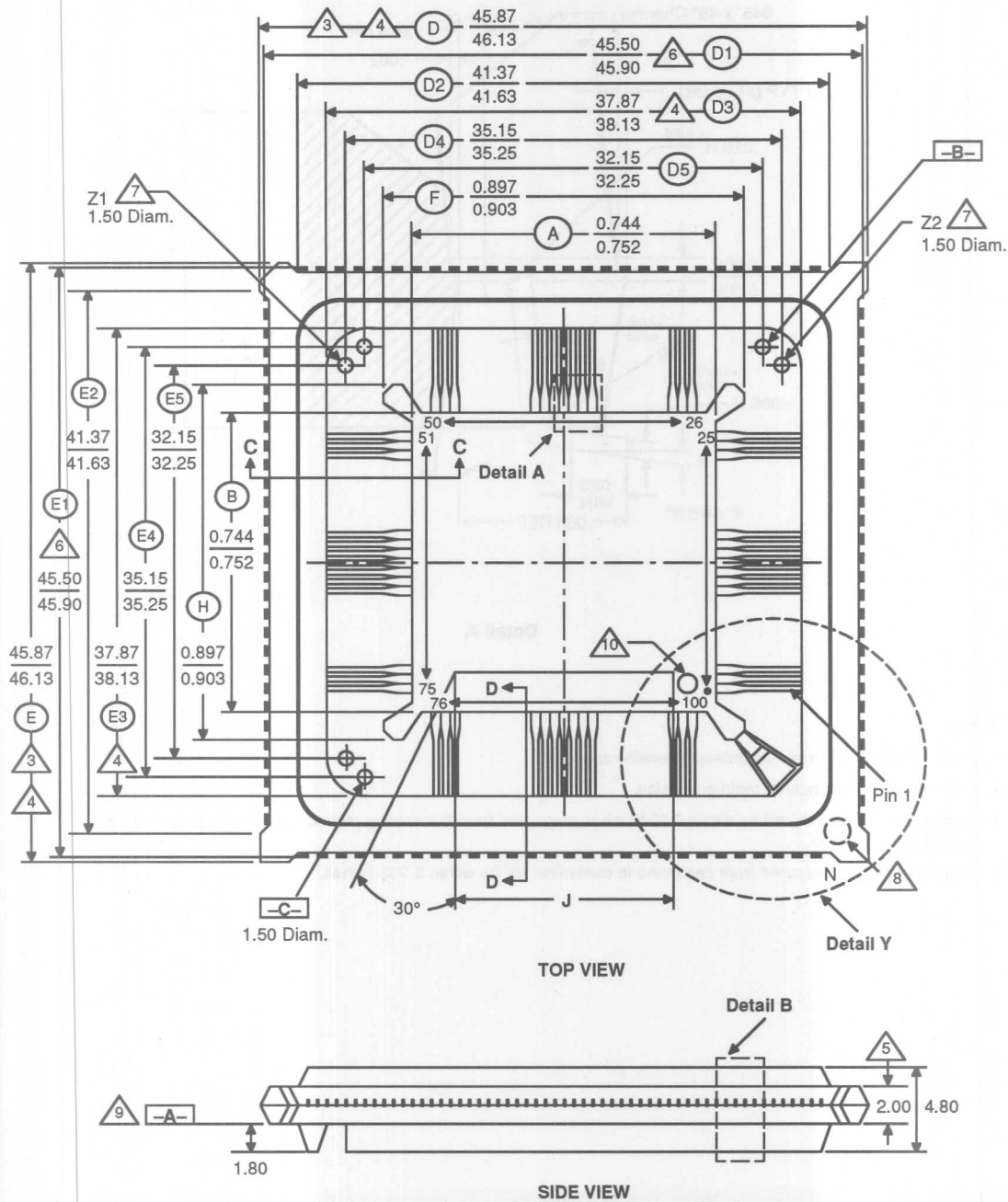
Detail A

## Notes:

1. All dimensions are in inches unless otherwise specified.
2. Dimensions do not include mold protrusion.
3. Coplanarity of all leads will be within 0.004 inches measured from the seating plan. Coplanarity is measured per specification 06-500.
4. Lead spacing as measured from centerline to centerline will be within 0.003 inches.

### PHYSICAL DIMENSIONS (continued)

### PQB-100 (Plastic Quad Flat Pack with Bumper, Molded Carrier Ring)

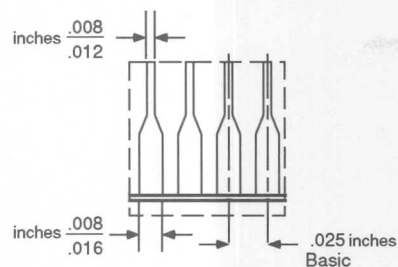


15680D  
CB 50  
4/8/92 SG

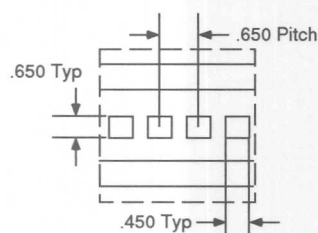


## PHYSICAL DIMENSIONS (continued)

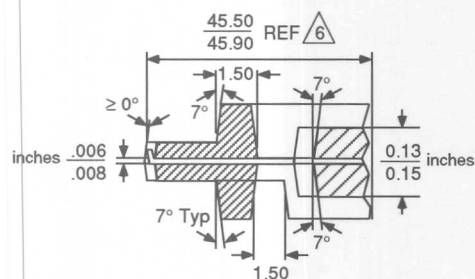
## PQB 100—Plastic Quad Flat Pack with Bumper, Molded Carrier Ring (continued)



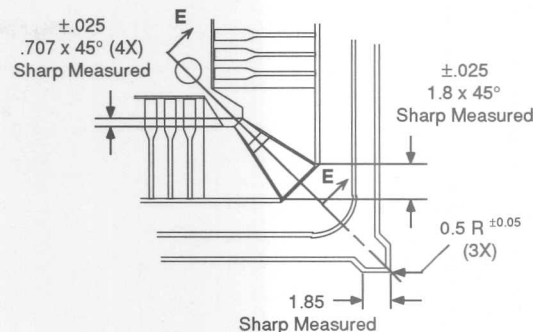
Detail A



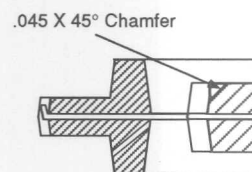
Detail B



Section C-C



Detail Y



Section D-D

## Notes:

1. All dimensions and tolerances conform to ANSI Y14.5M-1982.
2. Controlling dimensions: package is measured in inches and ring is measured in millimeters.
3. D and E dimensions do not include mold protrusion. Allowable mold protrusion is 0.2 mm per side.
4. D, D3, E, and E3 dimensions include mold mismatch and are measured at the parting line.
5. Dimensions are centered about centerline of lead material.
6. Dimensions D1 and E1 are from outside edge to outside edge of the test points.
7. There are six locating holes in the ring, -B- and -C- datum holes are used for trim form and excise of the molded package only. Holes Z1 and Z2 are used for electrical testing only.
8. This area is reserved for vacuum pickup on each of the four corners of the ring and must be flat within .025 mm. No ejector pins in this area.
9. Surface A is used for seating in socket applications.
10. Pin one orientation with respect to carrier ring as indicated.



# **Am386DXLV and Am386SXLV Microprocessors**

## **Technical Reference Manual**

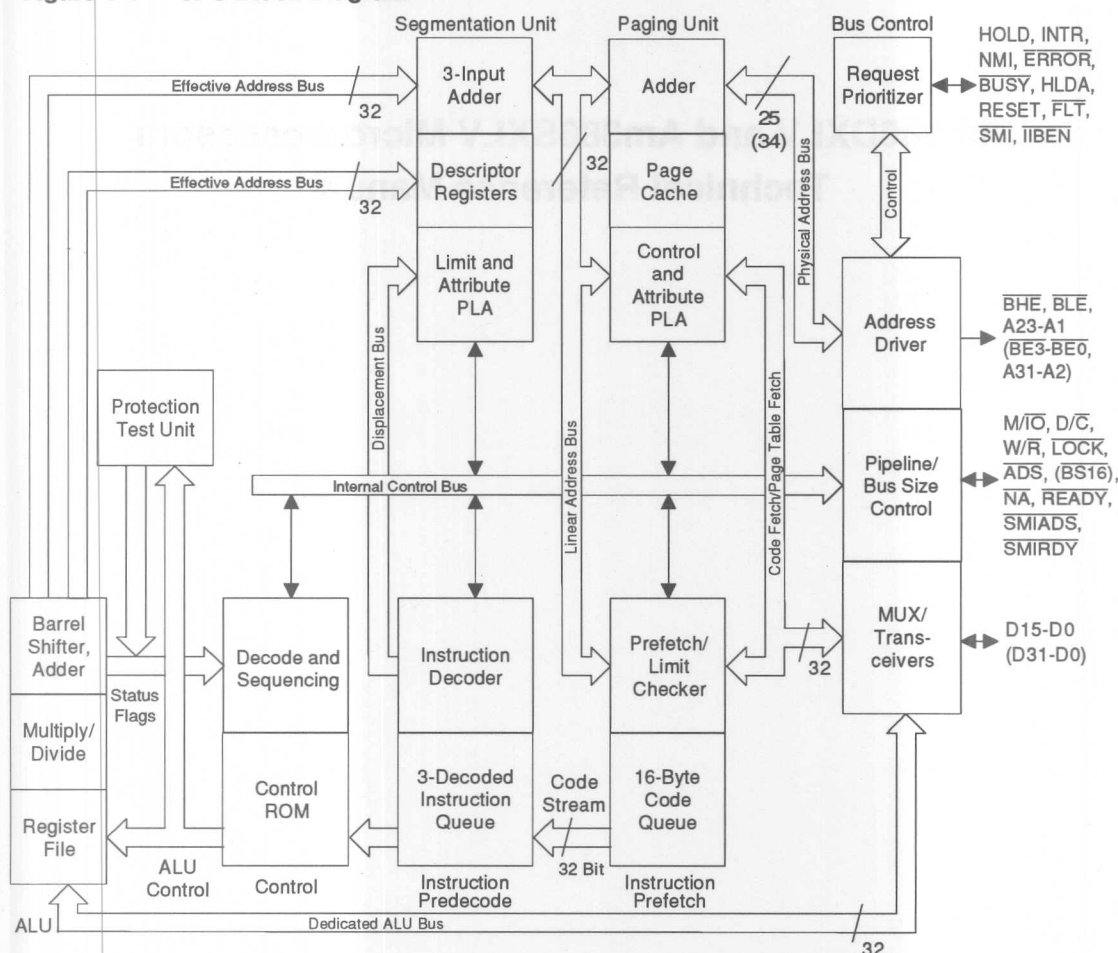
# INTRODUCTION



## GENERAL DESCRIPTION

The Am386DXLV and Am386SXLV microprocessors are low voltage, true static derivatives of the Am386 microprocessors with System Management Mode support. These microprocessors are ideal for systems where battery life and system weight are major concerns. The System Management Mode allows such systems very flexible control over system peripherals.

**Figure 1-1 CPU Block Diagram**



Note: DX specific characteristics shown in ().

16944A-001



## Low Voltage Features

The Am386DXLV and Am386SXLV microprocessors' lower operating voltage and true static design enable longer battery life and/or lower weight for portable systems. Lowering the operating voltage from 5.0 V to 3.3 V effectively halves CPU and core logic power consumption. Standby Mode of these microprocessors allows the clock frequency to be reduced to 0 MHz (DC) and retain full register contents. In Standby Mode, typical current draw is drastically reduced.

## System Management Mode (SMM)

SMM is an advanced power management feature that allows interruption and resumption of normal CPU activity. These interrupts and resumes allow the system designer to implement very effective power management schemes. This mode of operation is implemented with an SMI interrupt (System Management Interrupt), new SMM related CPU instructions, and an IIBEN enable for an I/O trapping feature (I/O Instruction Break Enable). SMI is a non-maskable, higher priority interrupt than NMI and has its own separate code space. SMI can be coupled with the I/O Instruction Break feature to implement transparent power management of peripherals.

## System Impact Summary

The Am386DXLV and Am386SXLV microprocessors embody the enabling technology for advanced portable systems. The features of low voltage, true static operation and SMM provide the system designer with new opportunities for innovation and system differentiation. The Am386DXLV and Am386SXLV CPU's are the ideal core for the next generation high-performance portable designs.



## SYSTEM MANAGEMENT MODE (SMM)

### SMM OVERVIEW

#### Capabilities

SMM provides a mechanism that interrupts the processor operation and resumes the interrupted operation transparent to the operation system or application being run on the system. This service routine resides in its own System Management address space. Therefore, supporting logic can be designed to allow arbitrary interruption of the processor's activity to allow for the execution of other code tasks. One obvious application of SMM is portable system power management.

Due to the pipelined nature of the Am386DXLV and Am386SXLV processors, accurately interrupting processor code execution with single instruction granularity is normally very difficult. This difficulty is primarily a concern when trapping I/O instructions. For this reason SMM is implemented in the Am386DXLV and Am386SXLV microprocessors with I/O Instruction Break capability. This feature allows I/O instruction trapping implementation.

#### Pin Descriptions

The CPU interface for SMM consists of four pins dedicated to the SMM function. One pin,  $\overline{\text{SMI}}$ , is the System Management Interrupt input. Two pins,  $\overline{\text{SMIADS}}$  and  $\overline{\text{SMIRDY}}$ , provide the control signals necessary for the separate SMI-mode memory space. The final pin,  $\overline{\text{IIBEN}}$ , allows for the enabling of non-pipelined I/O accesses to allow for I/O instruction trapping. The pin functions are defined as follows:

|                            |  |
|----------------------------|--|
| $\overline{\text{SMI}}$    | <b>System Management Interrupt</b> —Active Low I/O pin with active pull-up. This pin is the highest level interrupt input to the CPU.  |
| $\overline{\text{SMIADS}}$ | <b>SMI Address Status</b> —Active Low three-state output pin. When active, this pin indicates that a valid bus cycle to the separate SMM memory space has begun. It also validates the values on the $\overline{\text{W/R}}$ , $\overline{\text{D/C}}$ , $\overline{\text{M/I/O}}$ , $\overline{\text{BE3}}\text{--}\overline{\text{BE0}}$ , and A31–A2 pins (A23–A1, $\overline{\text{BHE}}$ , and $\overline{\text{BLE}}$ in the case of the Am386SXLV CPU). |
| $\overline{\text{SMIRDY}}$ | <b>SMI Ready</b> —Active Low input pin. This input terminates the current bus cycle to the $\overline{\text{SMIADS}}$ initiated accesses in the same manner the $\overline{\text{READY}}$ pin does for the $\overline{\text{ADS}}$ initiated address accesses.   |
| $\overline{\text{IIBEN}}$  | <b>I/O Instruction Break Enable</b> —Active Low input pin with active pull-up. When active, this pin enables the I/O instruction break feature. This feature disables execution pipelining for I/O instructions. If inactive, the feature is disabled and I/O cycles execute in a manner clock-for-clock compatible with the Am386DXL or Am386SXL processors.  |



## Pin Locations

The four SMM interface pins are at locations identified as No Connects (NCs) in the Am386DX/DXL and Am386SX/SXL microprocessor products. The pin locations for the Plastic Quad Flat Pack (PQFP) packages are shown in the tables below. The pin locations of all other signals remain the same as the standard products. This includes the no-connect pins not taken up by the new SMM signals. All no-connect pins are reserved for future use.

The Am386DXLV and Am386SXLV microprocessors are available in PQFP packages. The SMM related pin locations are noted in Table 2-1 (for the Am386DXLV microprocessor), and Table 2-2 (for the Am386SXLV microprocessor).

**Table 2-1 Am386DXLV CPU SMM Pin Locations**

| Name                       | PQFP Pin |
|----------------------------|----------|
| $\overline{\text{SMI}}$    | 59       |
| $\overline{\text{SMIADS}}$ | 37       |
| $\overline{\text{SMIRDY}}$ | 36       |
| $\overline{\text{IIBEN}}$  | 58       |

**Table 2-2 Am386SXLV CPU SMM Pin Locations**

| Name                       | PQFP Pin |
|----------------------------|----------|
| $\overline{\text{SMI}}$    | 43       |
| $\overline{\text{SMIADS}}$ | 31       |
| $\overline{\text{SMIRDY}}$ | 30       |
| $\overline{\text{IIBEN}}$  | 29       |

## Features

### SYSTEM MANAGEMENT MODE

SMM is implemented through a high priority System Management Interrupt (SMI). SMI is non-maskable and higher in priority than NMI. An SMM-based system can be implemented with the Am386DXLV and Am386SXLV processors' special SMI interface pins. This interrupt method can be used to perform system management functions independent of processor operating mode (Real, Protected, or Virtual 8086 modes).



Activating the SMI invokes a sequence that saves the operating state of the processor into a separate SMM memory address space, independent of the main system memory. After the state is saved, the processor is forced into Real Mode and begins execution out of that separate address space at the processor reset address (address FFFFFFF0h for the Am386DXLV microprocessor, address FFFF0h for the Am386SXLV microprocessor) where a jump to the SMM code is executed. This code performs its system management function and then resume execution of the normal system software by executing an SMM CPU state restore opcode sequence, which reloads the saved processor state and resume execution out of the main system memory space.

### I/O TRAPPING

I/O trapping can be implemented with the I/O Instruction Break feature. I/O trapping allows the system to turn off peripherals when they are not needed. The I/O trapping hardware can then intercept accesses to these peripherals by interrupting the instruction stream, turning on the peripheral, and then re-executing the trapped I/O instruction. When the signal  $\overline{IBEN}$  is active, the processor execution unit pauses upon execution of an I/O transfer until the end of the transfer cycle without affecting memory or register-related instructions. This pause allows non-pipelined interruption of the instruction via  $\overline{SMI}$ . After the SMM routine is complete, the I/O instruction can be re-executed and normal execution resumed. The I/O Instruction Break Enable signal is dynamic and can be enabled and disabled as the situation warrants. Thus, a system using this feature need only pause on I/O instructions when necessary.

### SMM Instructions

There are three specific instructions for SMM operation: SMI, UMOV, and RES3.

#### SMI—CALL TO SYSTEM MANAGEMENT INTERRUPT PROCEDURE

| Opcode | Instruction | Clocks           | Description                                   |
|--------|-------------|------------------|---|
| F1     | SMI         | DX-325<br>SX-357 | Call to System Management Interrupt Procedure |

|                       |   |
|-----------------------|---|
| <b>Operation</b>      | IF SMIE = 1<br>THEN ENTER SMM<br>ELSE<br>enter DEBUG exception routine (INT 1)<br>END   |
| <b>Description</b>    | When SMIE bit is set in Debug Control Register (DR7, bit 12) SMI forces the processor into SMM. If the SMIE bit is not set, the CPU executes an INT1 debug exception. |
| <b>Flags Affected</b> | SMMS bit in the Debug Status Register is set (DR6, bit 12 = 1).   |

**UMOV—MOVE DATA TO MAIN (USER) MEMORY**

| Opcode   | Instruction    | Clocks | Description                                     |
|----------|----------------|--------|---|
| 0F 10 /r | UMOV r/m8,r8   | 2/2    | Move byte register to r/m byte                  |
| 0F 11 /r | UMOV r/m16,r16 | 2/2    | Move word register to r/m word                  |
|          | UMOV r/m32,r32 | 2/2    | Move dword register to r/m dword                |
| 0F 12 /r | UMOV r8,r/m8   | 2/4    | Move r/m byte in main memory to byte register   |
| 0F 13 /r | UMOV r16,r/m16 | 2/4    | Move r/m word in main memory to word register   |
|          | UMOV r32,r/m32 | 2/4    | Move r/m dword in main memory to dword register |

Note: /r indicates the ModR/M byte of the instruction contains both a register and an r/m operand. Clock counts may be greater if access is misaligned.

**Operation** DEST ← SRC;

**Description** UMOV copies the second operand to the first operand with all memory references being initiated with ADS. UMOV functions like the MOV instruction with the exception that UMOV is an SMM instruction that references the normal memory system.

**Flags Affected** None

**RES3—RESTORE ALL CPU REGISTERS**

| Opcode | Instruction | Clocks | Description                           |
|--------|-------------|--------|---------------------------------------|
| 0F 07  | RES3        | DX-291 | Restore all CPU registers from memory |
|        |             | SX-366 |                                       |

**Operation** CPU Registers ← ES:[EDI]

**Description** RES3 loads all CPU registers from a memory-based table referenced by ES:[EDI]. When in SMM, the CPU save state table is loaded with SMIADS initiated cycles. When not in SMM, the CPU save state table is loaded with ADS initiated cycles. A new execution context will be established in this process. Upon completion of the instruction, the CPU resumes execution of the instruction stream described by the CPU save state table. The function of the RES3 opcode may change for future processors.

**Flags Affected** All flags are loaded from the CPU save state table.

## SMM FUNCTIONAL DESCRIPTION

### Introduction

The execution of an SMI has four distinct phases: the initiation of the mode via an active  $\overline{\text{SMI}}$  signal, a processor state save, execution of the SMM interrupt code, and a processor state restore (to resume normal operation). This process is shown in Figure 2-1.

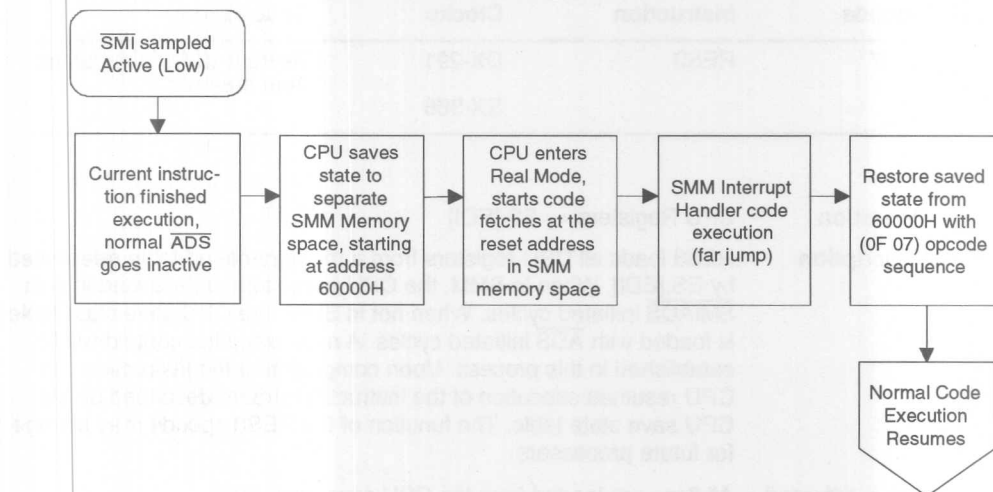
### SMM Initiation

SMM is initiated by driving  $\overline{\text{SMI}}$  active until the first active  $\overline{\text{SMIADS}}$  pulse. The CPU drives the  $\overline{\text{SMI}}$  pin active after the completion of the current operation. The active drive of the pin by the CPU is released at the end of the interrupt routine, following the last memory read of the stored save state.

An SMI cannot be masked by the CPU and is always recognized by the CPU, regardless of operating mode. This includes the Real, Protected, and Virtual 86 Modes of the processor.

SMI is the highest level external interrupt with precedence over both the NMI and INTR interrupts. SMIs cannot be nested. Another SMI request is not recognized until the completion of an SMI handler, when the CPU releases its drive of the SMI pin. Also upon entering SMM, NMI and INTR interrupts are not enabled. NMI requests are latched so processing of NMI's can be deferred until the SMM routine completes. INTR requests can be enabled by the EFLAGS register or the STI instruction. In the event of an exception, software interrupt (INT N or INT0), or INTR interrupt requests, the execution of the associated IRET instruction enables the recognition of pending NMI requests. In this case SMM software needs to comprehend the handling of NMI as well as standard INTR interrupts.

**Figure 2-1 Complete SMM Execution Sequence**



16944A-02

SMM can also be entered by setting the SMIE bit in debug register 7 and executing the soft SMI instruction. This instruction is the opcode 0F1h. Upon execution of this instruction, the system enters SMM as if a hardware SMI were detected.

### Processor State Save

The first set of SMM bus transfer cycles after the CPU's recognition of an active  $\overline{\text{SMI}}$  is the processor saving its state to an external memory in a separate address space from main system memory. There may be processor pre-fetch activity before entering SMM. The first SMM memory accesses are accomplished by using the  $\overline{\text{SMIADS}}$  and  $\overline{\text{SMIRDY}}$  pins for initiation and termination of bus cycles (instead of the  $\overline{\text{ADS}}$  and  $\overline{\text{READY}}$  pins). The addresses to which the CPU saves its state are 60000h–600CBh and 60100h–60127h. These are fixed address locations for each register saved. Note that each 32-bit register saved results in two 16-bit transfer cycles on the Am386SXLV CPU. A map of the save/restore state memory area is in Appendix A.

The state save transfer cycles are initiated with  $\overline{\text{SMIADS}}$ , and the bus cycle status pins indicate a memory data write:  $\text{W}/\overline{\text{R}}=1$ ,  $\text{D}/\overline{\text{C}}=1$ , and  $\text{M}/\overline{\text{IO}}=1$ . These cycles must be terminated with  $\overline{\text{SMIRDY}}$ . The Am386DXLV processor accesses only SMM addresses as 32-bit accesses, ignoring  $\overline{\text{BS16}}$  for SMM cycles.

A total of 114 data transfer cycles are required for the Am386SXLV CPU and 61 cycles for the Am386DXLV CPU to complete the save state operation: there are fifty-three 32-bit registers and eight 16-bit registers to be saved.

### SMI Code Execution

After the processor state is saved to the separate SMM memory space, the execution of the interrupt routine code begins. The processor enters Real Mode and begins fetching code from the reset address in the separate SMM memory space. The processor state and register contents upon entering SMM are detailed in Appendix A. Typically, the first thing the interrupt routine code does is a jump to the Real-mode entry point for the SMM interrupt routine, which is also in SMM memory space.

The code fetch bus transfer cycles are initiated with  $\overline{\text{SMIADS}}$ , and the bus cycle status pins indicate an instruction fetch:  $\text{W}/\overline{\text{R}}=0$ ,  $\text{D}/\overline{\text{C}}=0$ , and  $\text{M}/\overline{\text{IO}}=1$ . These cycles must be terminated with  $\overline{\text{SMIRDY}}$ . The Am386DXLV processor accesses only SMM addresses as 32-bit accesses, ignoring  $\overline{\text{BS16}}$  for SMM cycles.

Any interrupt routine code can be executed within SMM. The SMM code can be located anywhere within the SMM address space, except for where the processor state is saved. If Protected Mode is enabled, SMM interrupt code has full access to the 4-Gb address space. I/O bus cycles, as a result of the IN, OUT, INS, and OUTS instructions, are directed to the normal address space, utilizing the normal  $\overline{\text{ADS}}$  and  $\overline{\text{READY}}$  bus interface signals. This facilitates code that manipulates system hardware registers through the standard I/O subsystem. A separate I/O space does not need to be implemented.

### Processor State Restore

Returning to normal code execution in the main system memory, including restoring the processor operating mode, is accomplished by executing a CPU restore instruction RES3 (opcode 0Fh 07h). This code invokes a restore CPU state operation that reloads the CPU registers from the saved data in the memory space controlled by  $\overline{\text{SMIADS}}$  and  $\overline{\text{SMIRDY}}$  signals.

For the RES3 instruction to execute properly, the ES:EDI register pair must point to physical address 60000h where the interrupted CPU state is saved. The execution can

be accomplished in Real Mode by loading ES with 6000h and clearing the EDI register. Then RES3 should be executed to start the restore state operation. After completion of the restore state operation, the  $\overline{\text{SMI}}$  pin is deactivated by the CPU and normal code execution continues at the address specified by the contents of the CS:EIP register in the processor save state.

The state restore transfer cycles are initiated with  $\overline{\text{SMIADS}}$ , and the bus cycle status pins indicate a memory data read:  $\text{W}/\overline{\text{R}}=0$ ,  $\text{D}/\overline{\text{C}}=1$ , and  $\text{M}/\overline{\text{IO}}=1$ . These cycles must be terminated with  $\overline{\text{SMIRDY}}$ . There are 114 data transfer cycles required for the Am386SXLV processor and 61 cycles required for the Am386DXLV processor to complete the processor state restore. The Am386DXLV processor will only access SMM addresses as 32-bit accesses, ignoring  $\overline{\text{BS16}}$  for SMM cycles.

If the CPU was powered down for power management purposes while in SMM, the RES3 instruction can be executed in Normal Mode after power up. This execution can be done if the system logic provides normal access to the saved CPU state in SMM memory space. The ES:EDI register pair should point to the Normal Mode address of the saved state and the RES3 opcode (0Fh 07h) can be executed. In this case the CPU executes the restore state with  $\overline{\text{ADS}}$  initiated cycles.

## I/O INSTRUCTION BREAK

### Functional Description

The operation of the I/O Instruction Break feature allows external system logic to interrupt the processor on I/O instruction boundaries. This interruption is achieved by the system logic driving  $\overline{\text{IIBEN}}$  active and providing logic to assert  $\overline{\text{SMI}}$  after recognition of an I/O bus cycle. This causes the CPU to enter SMM immediately after executing the I/O instruction if  $\overline{\text{SMI}}$  was asserted.

Internally, I/O Instruction Break pauses normal execution at the end of the I/O instruction but before execution of the following instruction. Assertion of  $\overline{\text{IIBEN}}$  causes the CPU execution unit to pause on an I/O bus cycle until it receives the final  $\overline{\text{READY}}$  associated with the I/O request. In the case of misaligned requests from the execution unit, more than one physical I/O bus cycle can occur. After the final  $\overline{\text{READY}}$  is received, the I/O instruction is completed.

When an I/O transfer that needs to be trapped is issued with  $\overline{\text{IIBEN}}$  active, the system logic must assert  $\overline{\text{SMI}}$  for more than three CLK2 periods before asserting  $\overline{\text{READY}}$  to terminate the cycle. The CPU immediately goes into SMM without executing the next instruction. Failure to meet the required setup time results in a standard SMM entry at some point after the execution of the I/O instruction (the I/O instruction will not be trapped).

The execution of coprocessor cycles is not affected, therefore coprocessor cycles cannot be trapped as other I/O cycles.

The "I/O breakable" instructions are listed below. With I/O Instruction Break enabled, the execution clock count for these instructions increases. However, coprocessor I/O instructions executions do not slow down. No memory or register instruction execution times are affected.

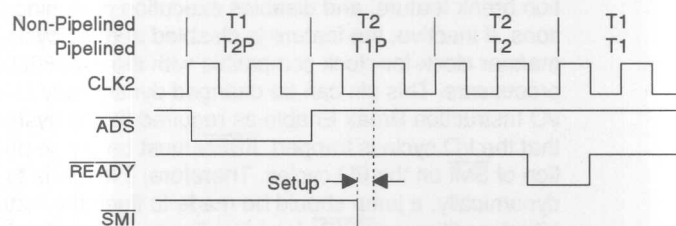
| Instruction | Opcodes        |
|-------------|----------------|
| IN          | E4, E5, EC, ED |
| OUT         | E6, E7, EE, EF |
| INS         | 6C, 6D         |
| OUTS        | 6E, 6F         |
| REP INS     | F3 6C, F3 6D   |
| REP OUTS    | F3 6E, F3 6F   |

For the I/O string instructions (with or without the REP prefix), the assertion of  $\overline{\text{SMI}}$  marks the last executed I/O transfer pair. With IIBEN active, an  $\overline{\text{SMI}}$  asserted on the first I/O read or write does not usually result in another following I/O transfer cycle until after the  $\overline{\text{SMI}}$  code is executed. An  $\overline{\text{SMI}}$  asserted on the I/O read part of a REP INS instruction has a corresponding memory write cycle before entering SMM. The one exception to this function is the REP OUTS instruction. The operation of REP OUTS can, in some cases, execute two memory read-I/O write pairs before trapping depending upon the state of the CPU when  $\overline{\text{SMI}}$  is asserted. In this case an overrun flag is set in the CPU save state so the need for corrective action can be detected by the SMM routine.

The I/O Instruction Break feature is implemented such that asserting  $\overline{\text{SMI}}$  properly traps an I/O Instruction if the  $\overline{\text{SMI}}$  pin is sampled active (Low) three CLK2 edges before the CLK2 edge that ends the I/O cycle with an active  $\overline{\text{READY}}$  pin. Therefore, to implement I/O instruction break for a particular I/O device, the access for that device must be at least a 1 CPU wait state cycle. This timing applies to both non-pipelined and pipelined cycles. (see Figure 2-2). Note that  $\overline{\text{SMI}}$  is set up to the rising edge of CLK2 on the beginning of the second phase of the processor state.

Depending on the state of the pre-fetch queue at the time  $\overline{\text{SMI}}$  is asserted, instruction fetch cycles might occur on the normal  $\overline{\text{ADS}}$  interface before the SMM save state begins with the assertion of  $\overline{\text{SMIADS}}$ . However, the pre-fetched code is not executed.

**Figure 2-2 I/O Instruction Break  $\overline{\text{SMI}}$  Timing**



16944A-03



## SMM HARDWARE INTERFACE

### Pin Behavior

#### S<sub>M</sub>I

System Management Interrupt is an active Low I/O pin with an active pull-up. This pin is the highest level interrupt input to the CPU. In order to initiate SMM, S<sub>M</sub>I should be driven Low by an open collector driver until the first S<sub>M</sub>IADS pulse occurs to ensure recognition. The signal should then be released. In input mode, S<sub>M</sub>I must be synchronous to CLK2. When the interrupt is recognized and service has begun, the CPU drives S<sub>M</sub>I active to indicate that the SMI service routine is executing. The CPU drives S<sub>M</sub>I inactive and then releases it to the internal CPU pull-up upon completing the CPU restore state process. The CPU drives S<sub>M</sub>I High for two CLK2 periods and then releases S<sub>M</sub>I to the internal weak pull-up after the saved state is reloaded. The S<sub>M</sub>I pull-up is active during RESET and whenever the signal is not driven active by the CPU. The pull-up is disabled when the CPU is driving S<sub>M</sub>I to minimize CPU power consumption. Note that S<sub>M</sub>I is not floated during HOLD states while in SMM.

#### S<sub>M</sub>IADS

SMI Address Status is an active Low three-state output pin. When active, this pin indicates that a valid bus cycle to the separate SMI memory space has begun. S<sub>M</sub>IADS also validates the values on the W/R, D/C, M/I/O, BE<sub>3</sub>–BE<sub>0</sub>, and A31–A2 pins (A23–A1, BHE, and BLE in the case of the Am386SXLV CPU). The function of S<sub>M</sub>IADS is analogous to ADS. This is a three-state output which floats during a bus HOLD cycle, as indicated by an active HLDA pin.

#### S<sub>M</sub>IRDY

SMI Ready is an active Low input pin with an internal pull-up. This input terminates the current bus cycle to S<sub>M</sub>IADS initiated accesses in the same manner that the READY pin terminates ADS initiated accesses. The signal S<sub>M</sub>IRDY is a CPU input that is synchronous to CLK2. S<sub>M</sub>IRDY's function is analogous to READY; however, S<sub>M</sub>IRDY and READY may not be sourced from the same signal.

#### IIBEN

I/O Instruction Break Enable is an active Low input pin with an active pull-up. IIBEN is an asynchronous CPU input that is internally synchronized by the CPU to CLK2 and must be valid for several CLK2 pulses to be recognized. When active, this pin enables the I/O instruction break feature, and disables execution pipelining for I/O instructions. If inactive, the feature is disabled and I/O cycles execute in a manner clock-for-clock compatible with the Am386DXL or Am386SXL processors. This pin can be changed dynamically to enable or disable I/O Instruction Break Enable as required by the system. To ensure that the I/O cycle is trapped, IIBEN must be active prior to the assertion of S<sub>M</sub>I on the I/O cycles. Therefore, if IIBEN is to be changed dynamically, a jump should be made to flush the instruction queue after transitioning IIBEN from inactive to active. The IIBEN pull-up is active during RESET and whenever the signal is not driven active by the system. Once the CPU detects IIBEN being driven by the system, the active pull-up is disabled until the next active RESET pulse or until IIBEN is sampled High. Therefore, IIBEN must be driven at all times if the I/O Break Enable feature is used. In systems not using the I/O Instruction Break feature, the pin does not need to be connected.



## Bus Cycles

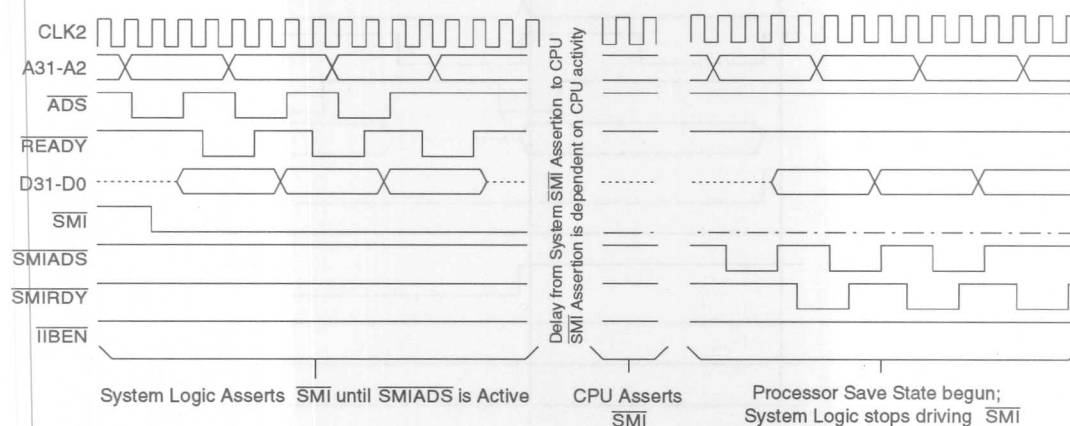
Bus cycles for the CPU during the SMM sequence are much like conventional CPU cycles. SMM memory transactions are requested and terminated with the SMM specific control lines ( $\overline{\text{SMIADS}}$ ,  $\overline{\text{SMIRDY}}$ ). Processor Save State and SMM code execution always occur with the SMM specific control lines. I/O transactions take place in the standard CPU I/O space and therefore use standard control lines ( $\overline{\text{ADS}}$ ,  $\overline{\text{READY}}$ ) to request and terminate the cycles. Special cycles such as interrupt acknowledge, HALT, and shut down cycles are also indicated by activity on the standard control lines.

The state of  $\overline{\text{NA}}$  is ignored during SMM code fetches because only non-pipelined cycles can be generated in SMM. Because of this, the standard memory system should not be allowed to perform pipelined cycles while  $\overline{\text{SMI}}$  is active. Due to pipelined  $\overline{\text{ADS}}$  timing, memory transfers to/from non-pipelined SMM and a pipelined standard memory could have unpredictable results. Therefore,  $\overline{\text{NA}}$  must not be active when in SMM.  $\overline{\text{NA}}$  should be driven inactive at the receipt of the first  $\overline{\text{SMIADS}}$  and held inactive until the processor drives  $\overline{\text{SMI}}$  inactive.

All SMM address space cycles made by the Am386DXLV processor are 32-bit accesses. The 32-bit processor ignores  $\text{BS16}$  during cycles directed to the SMM address space. The Am386DXLV processor recognizes  $\text{BS16}$  during accesses to the normal address space. While a double word address space must be implemented, it should be noted that word and byte accesses, as required by instructions, function properly utilizing the byte enables.

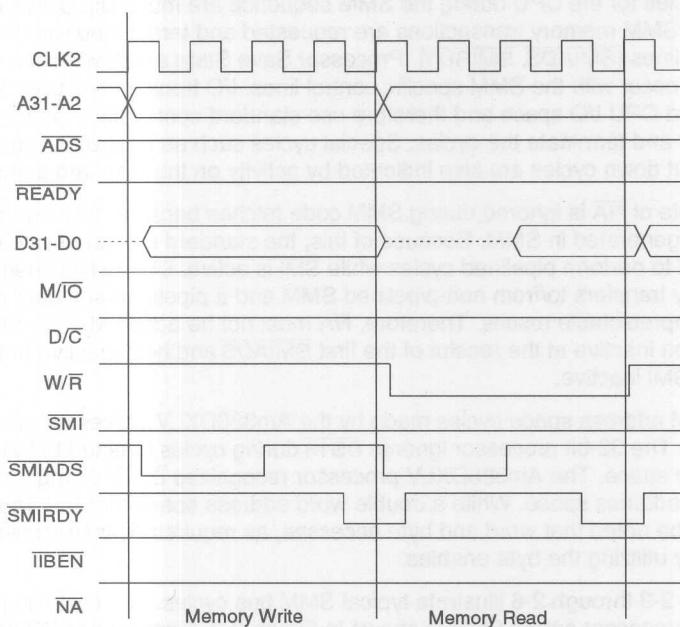
Figures 2-3 through 2-6 illustrate typical SMM bus cycles. The cycle sequence executed as the processor enters SMM is shown in Figure 2-3. Memory and I/O access while in SMM are detailed in Figures 2-4 and 2-5. Access to standard system memory from SMM with the UMOV instruction is illustrated in Figure 2-6. Figure 2-7 shows the processor exiting SMM and resuming the interrupted instruction sequence.

**Figure 2-3  $\overline{\text{SMI}}$  Assertion, Start of CPU Save State**



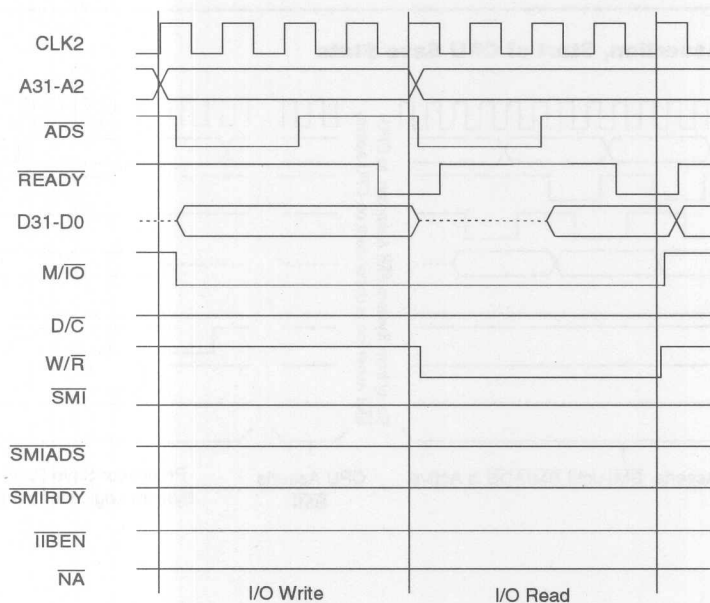
16944A-004

**Figure 2-4 SMM Memory Write and Read**



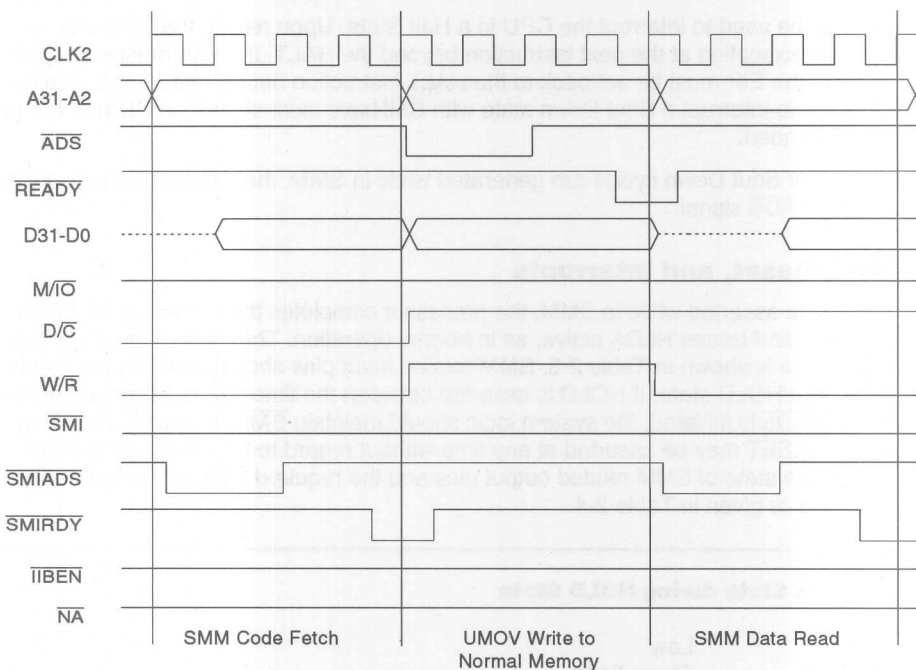
16944A-05

**Figure 2-5 SMM I/O Write and Read**



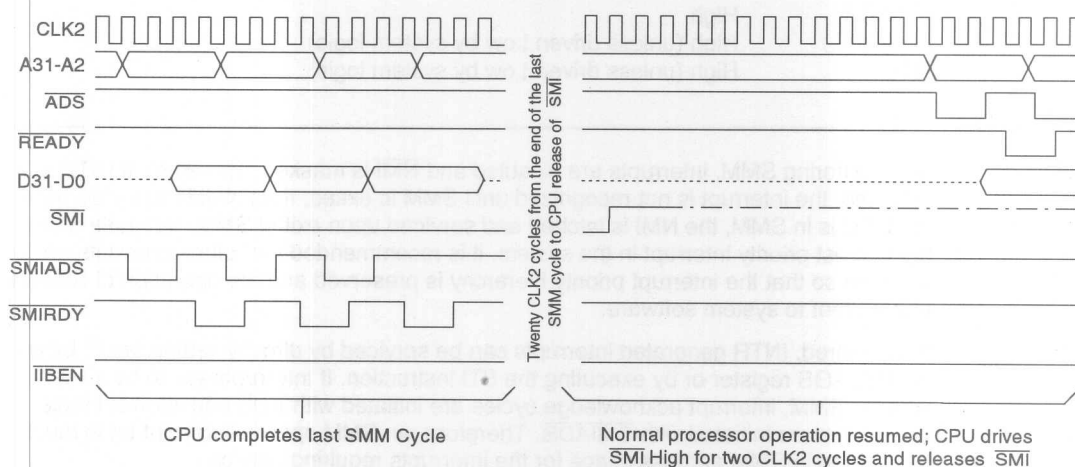
16944A-06

**Figure 2-6 SMM UMOV Example**



16944A-007

**Figure 2-7 End of CPU Restore State**



16944A-008

## SMI, Halt, and Shut Down

$\overline{\text{SMI}}$  can be used to interrupt the CPU in a Halt State. Upon return, the CPU will continue execution at the next instruction beyond the HALT. If SMM transparency is desired, the EIP must be set back to the HALT instruction before restore is executed. Attempts to interrupt a Shut Down state with  $\overline{\text{SMI}}$  have inconsistent results and are not recommended.

If HALT or Shut Down cycles are generated while in SMM, they are generated with the standard  $\overline{\text{ADS}}$  signal.

## Hold, Reset, and Interrupts

If HOLD is asserted while in SMM, the processor completes the current cycle, floats its bus pins, and issues HLDA active, as in normal operation. The state of the SMM related output pins is shown in Table 2-3. SMM related input pins should maintain their state during the HOLD state. If HOLD is asserted between the time  $\overline{\text{SMI}}$  is asserted and the first  $\overline{\text{SMIADS}}$  is initiated, the system logic should maintain  $\overline{\text{SMI}}$  active until it is recognized. RESET may be asserted at any time without regard to the processor's SMM state. The state of SMM related output pins and the required state of the SMM related input pins is given in Table 2-4.

**Table 2-3 SMM Pin State during HOLD State**

|                            |             |
|----------------------------|-------------|
| $\overline{\text{SMI}}$    | Low         |
| $\overline{\text{SMIADS}}$ | Three-State |

**Table 2-4 SMM Pin State during RESET**

|                            |   |
|----------------------------|---|
| $\overline{\text{SMI}}$    | Not driven if not in SMM, may be driven High for two CLK2 periods if in SMM |
| $\overline{\text{SMIADS}}$ | High  |
| $\overline{\text{SMIRDY}}$ | High (unless driven Low by system logic)                                    |
| $\overline{\text{IBEN}}$   | High (unless driven Low by system logic)                                    |

Upon entering SMM, interrupts are disabled and  $\overline{\text{NMI}}$  is masked. Therefore, if INTR is asserted, the interrupt is not recognized until SMM is exited. If an NMI is asserted while the CPU is in SMM, the NMI is latched and serviced upon exit of SMM. Because  $\overline{\text{SMI}}$  is the highest priority interrupt in the system, it is recommended that other interrupts not be serviced so that the interrupt priority hierarchy is preserved and the operation of SMM is transparent to system software.

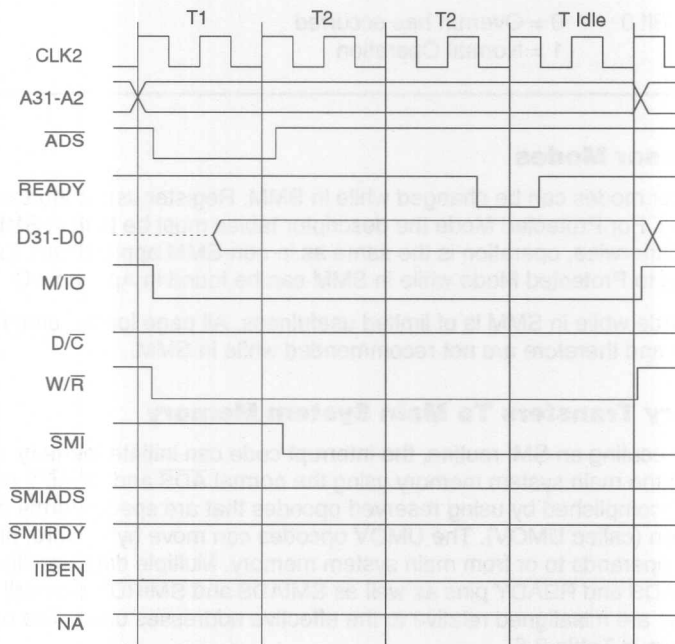
If so desired, INTR generated interrupts can be serviced by directly setting the IF bit in the EFLAGS register or by executing the STI instruction. If interrupts are to be serviced while in SMM, interrupt acknowledge cycles are initiated with  $\overline{\text{ADS}}$  and interrupt table references are initiated with  $\overline{\text{SMIADS}}$ . Therefore, an SMM specific interrupt table must be created in SMM address space for the interrupts requiring service.

When servicing interrupts or exceptions, the execution of the IRET instruction enables the recognition of NMI requests. One possible strategy for the SMM code is to disable NMI in external system logic, a common feature in PC architecture systems. Therefore, if interrupts or exceptions are serviced while in SMM, the system designer must account for all possible INTR and NMI requests.

### I/O Trapping

CPU cycles with I/O Instruction Break Enabled ( $\overline{\text{IIBEN}}$  active) are not physically different in terms of cycle definition. The cycle simply pauses the execution unit of the CPU until the bus unit can complete the requested I/O cycle. The resulting bus traffic is different than normal traffic only because there are more idle cycles than normal and code pre-fetch cycles may appear earlier relative to data cycles. Figure 2-8 illustrates a typical I/O instruction break. In Figure 2-8, no more instructions will be executed before the CPU enters SMM. However, Bus Unit pre-fetch cycles are possible before the CPU enters SMM.

**Figure 2-8 Typical I/O Instruction Break**



16944A-09

## SMM Software Features

There are several features of the SMM function that provide support for special operations during the execution of the system's software. These features involve the execution of reserved opcodes to invoke specific SMM related operations.

### Processor Save State

Upon entering SMM, the processor saves its entire state in the SMM address block from 60000h–60127h. The complete use of this space is detailed in Appendix A. One detail of the Save State which the SMM routine regularly checks in I/O trapping applications is the REP OUTS overrun flag. This overrun flag is contained in bit 0 of the location 6006Ch (see Table 2-5). If the overrun flag is set, no overrun has occurred. If the overrun flag is cleared, a REP OUTS overrun has occurred (that is, two pairs of memory-I/O accesses have occurred). In the case of an overrun, the SMM routine must alter the affected registers to replay two cycles instead of one in order to ensure proper re-execution of the trapped I/O instruction.

**Table 2-5 REP OUTS Overrun Flag**

|              |  |
|--------------|--|
| 6006Ch Bit 0 | 0 = Overrun has occurred<br>1 = Normal Operation |
|--------------|--|

### Processor Modes

Processor modes can be changed while in SMM. Register use is the same as non-SMM operation. For Protected Mode the descriptor tables must be built in SMM memory space. Otherwise, operation is the same as in non-SMM applications. Example code for changing to Protected Mode while in SMM can be found in Appendix C.

Page Mode while in SMM is of limited usefulness. All page loads come from the normal memory and therefore are not recommended while in SMM.

### Memory Transfers To Main System Memory

While executing an SMI routine, the interrupt code can initiate memory data reads and writes to the main system memory using the normal  $\overline{ADS}$  and  $\overline{READY}$  pins. This initiation is accomplished by using reserved opcodes that are special forms of the MOV instruction (called UMOV). The UMOV opcodes can move byte, word, or double word register operands to or from main system memory. Multiple data transfers using the normal  $\overline{ADS}$  and  $\overline{READY}$  pins as well as  $\overline{SMIADS}$  and  $\overline{SMIRDY}$  pins will occur if the operands are misaligned relative to the effective addresses used. The opcode formats are shown in Table 2-6.

The UMOV instruction can use any of the 386 addressing modes, as specified in the ModR/M byte of the opcode. Note that the 16- and 32-bit versions are the same opcodes with the exception of the 066h operand size prefix. Table 2-6 illustrates the UMOV opcode and encoded examples. Assembler macros examples can be found in Appendix B. The data in Table 2-6 assumes a default operand and address sizes of 16 bits. Using 32-bit Protected Mode changes the use of prefixes.

**Table 2-6 UMOV Opcode Form**

| Opcode       | Instruction    | Description                      | Encoded Example                 |
|--------------|----------------|----------------------------------|---------------------------------|
| 0F 10 ModR/M | UMOV r/m8,r8   | Move byte register to r/m byte   | 0F 10 00 ; umov [bx+si],al      |
| 0F 11 ModR/M | UMOV r/m16,r16 | Move word register to r/m word   | 0F 11 00 ; umov [bx+si],ax      |
| 0F 11 ModR/M | UMOV r/m32,r32 | Move dword register to r/m dword | 66 0F 11 00 ; umov [bx+si], eax |
| 0F 12 ModR/M | UMOV r8,r/m8   | Move r/m byte to byte register   | 0F 12 00 ; umov al,[bx+si]      |
| 0F 13 ModR/M | UMOV r16,r/m16 | Move r/m word to word register   | 0F 13 00 ; umov ax,[bx+si]      |
| 0F 13 ModR/M | UMOV r32,r/m32 | Move r/m dword to dword register | 66 0F 13 00 ; umov eax,[bx+si]  |

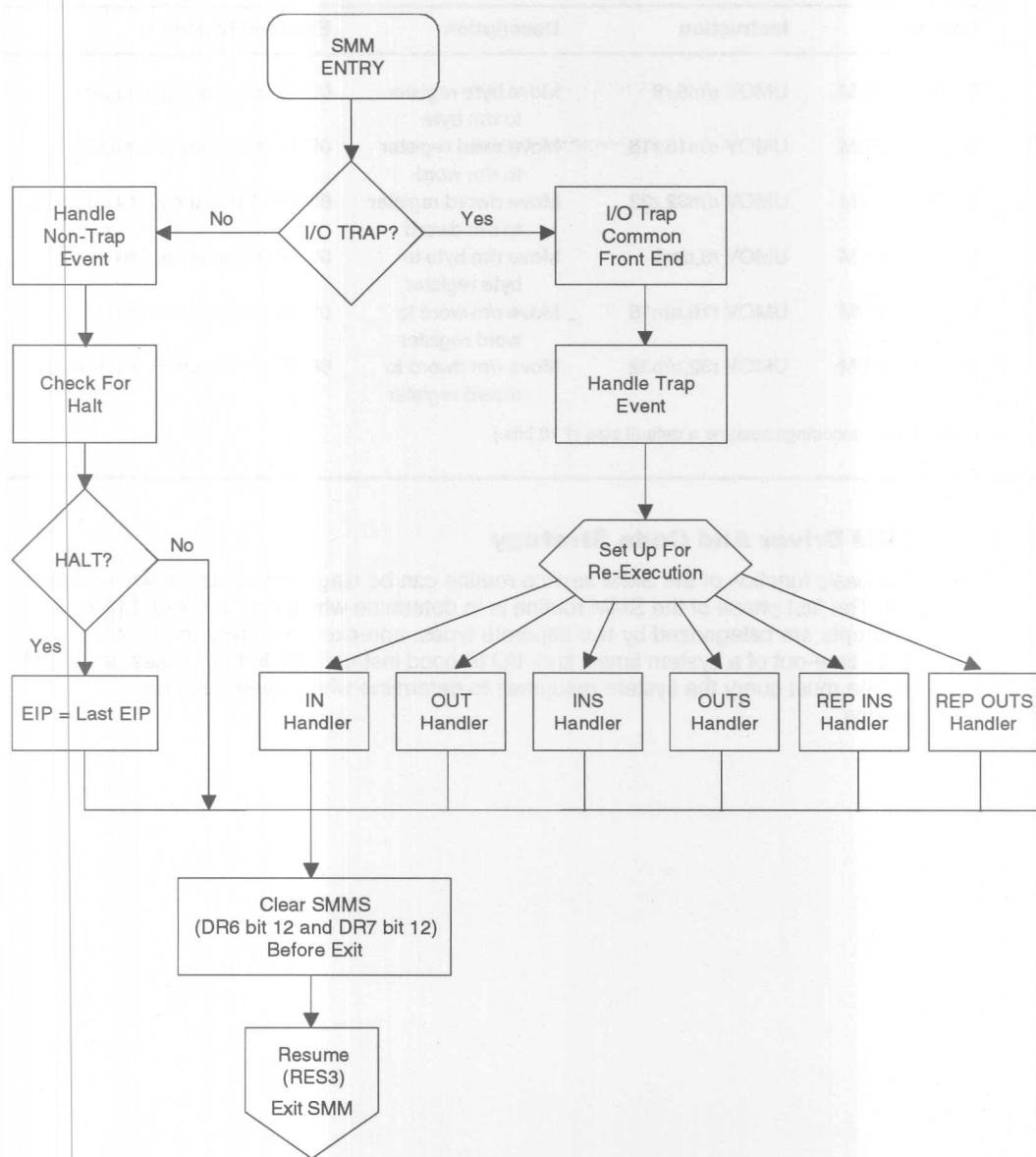
(Note: These encodings assume a default size of 16 bits.)

### SMM Driver And Code Strategy

The basic function of the SMM service routine can be diagrammed as shown in Figure 2-9. The first phase of the SMM routine is to determine why it was invoked. SMM interrupts are categorized by two separate types: non-execution system events (e.g., time-out of a system timer) and, I/O trapped instructions. In both cases, the SMM routine must query the system resources to determine which event requires service.



**Figure 2-9 SMM Service Routine Flow Chart**



16944A-10

The second phase of SMM code execution is the service phase. The path chosen on the flow chart depends on the determination of the SMI event source. The non-I/O trapped events are primarily system dependent with only one special instruction case to be handled. For I/O trapped events, it is assumed that the interrupted instruction requires re-execution to properly resume operation. This could be due to an I/O operation directed toward a peripheral which is powered down. In this case, the interrupted instruction must be examined and appropriate action taken to re-execute it properly.

Depending on the implementation of the system logic, multiple SMM interrupt sources may be pending. The SMM code may need to poll for any other possible sources before exiting the SMM routine.

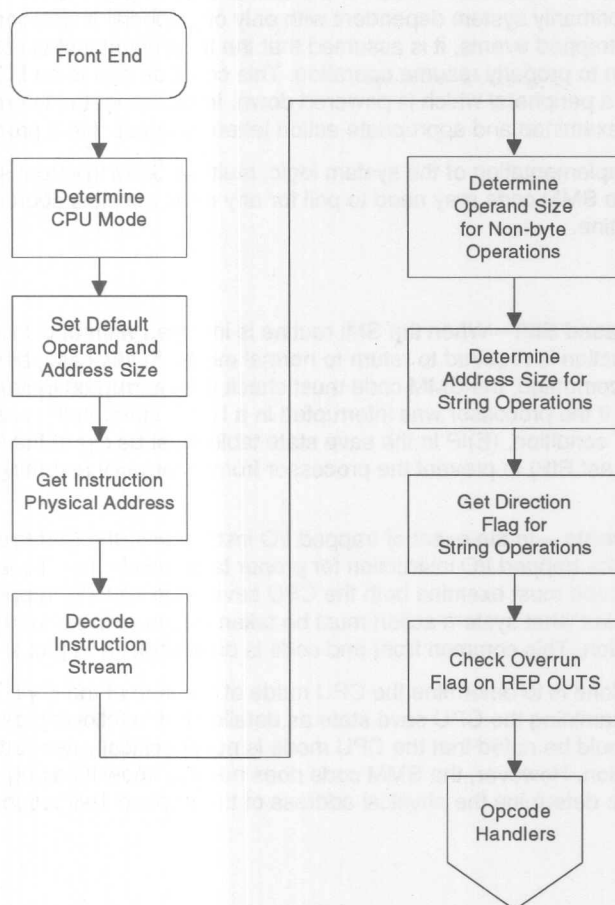
### Service Phase

**Non-I/O Trapped Based SMI**—When the SMI routine is involved without I/O trapping, only the RES3 instruction is required to return to normal mode. In this case, before issuing the resume command, the SMM code must check the interrupted instruction stream to determine if the processor was interrupted in a HALT state. In the case of an interrupted HALT condition, (E)IP in the save state table must be set to the HLT instruction (EIP <= Last EIP) to prevent the processor from incorrectly resuming at the next instruction.

**I/O Trapped SMI Events**—In the event of trapped I/O instructions, the SMM routine may need to replay the trapped I/O instruction for proper task resumption. To accomplish this, the SMM code must examine both the CPU save state and the trapped I/O instruction to determine what system action must be taken and to prepare for the I/O instruction re-execution. This common front end code is diagramed in Figure 2-10.

The first task to be done is to determine the CPU mode at the time of the SMI. This can be determined by examining the CPU save state as detailed in the following pseudo-code example. It should be noted that the CPU mode is not specifically needed for instruction re-execution. However, the SMM code does need to know if paging was enabled to be able to determine the physical address of the trapped instruction.

**Figure 2-10 I/O Trap Front End Flow Chart**



16944A-11

#### DETERMINING THE CPU MODE:

```

IF NOT PROTECTED (CRO Bit), Loc 60000H) THEN (*Note: it is not re-
  MODE = REAL;
ELSE
  IF NOT VM (EFLAG Bit 17, Loc 60004H) THEN
    MODE = PROTECT:
  ELSE
    MODE = VIRT86;
  ENDIF;
ENDIF;
IF PG=1 (CRO Bit 31, Loc 60000H) THEN
  PAGING = ON;
ELSE
  PAGING = OFF;
ENDIF;
  
```

(\*We need to know if paging was enabled\*)

(\*Note: it is not required to determine the exact mode of the processor to prepare for an instruction re-execution. This is shown to illustrate the process if other SMM code requires it.\*)

The next task is to set the default operand and address sizes to facilitate the proper decoding of the interrupted instruction's operand and address fields. This is accomplished by examining the code segment attributes from the CPU save state as shown in the following example.

#### SETTING DEFAULT SIZES:

```
IF (D = 0) (CS ATTR Bit 22, Loc 600B4H) THEN
    DEFOPERAND = 16;
    DEFADDR = 16;
ELSE
    DEFOPERAND = 32;
    DEFADDR = 32;
ENDIF;
```

To access the interrupted instruction, the SMM code must now get the instruction's physical address. This is a simple calculation if the CPU was not in Page Mode. If paging was enabled the situation is more complicated since the instruction's physical address must be calculated from information in the CPU save state and the normal memory-based page tables. Again, data from the CPU save state is used as shown in the following code.

#### GET THE INSTRUCTION'S PHYSICAL ADDRESS:

```
CSBASE = Loc 600B8H (32-bit value); (*valid for Real,
OFFSET = LASTEIP (Loc 60124H);      Protect, and V86 Modes*)

LINEARADDR = CSBASE + OFFSET;

IF PAGING = OFF THEN
    PHYSICALADDR = LINEARADDR;
ELSE
    DIRENTRYADDR = PDBR(Actual CR3 Value) + (LINEARADDR[31:22]) * 4;
    PAGEENTRYADDR = (DIRENTRY AND FFFFF000H) + (LINEARADDR[21:12]) * 4;
    PHYSICALADDR = (PAGEENTRY AND FFFFF000H) + (LINEARADDR[11:0]);
ENDIF;

MAXINSTRLLENGTH = 5 (bytes);          (*the longest I/O instruction is
                                         5 bytes w/o redundant overrides*)
```

Now that all the information needed to access the instruction has been extracted from the CPU save state, the instruction can be loaded into an array and decoded. This example assumes only required override bytes are used and there are no duplicate overrides. The order of override prefixes is assumed to be standard. Production code may want to take non-standard code into account. The maximum instruction length of a poorly formed instruction with redundant override prefixes is 15 bytes. Note that the pseudo-code does not show the detail of enabling Protected Mode for physical instruction addresses outside the Real Mode address space. An example of how to change to Protected Mode while in SMM can be found in Appendix C.

#### GET INSTRUCTION STREAM:

```
INSTR[0:4] = [PHYSICALADDR : PHYSICALADDR + 4] (*use Protected Mode if
                                                Address > 1M*)
```

## DECODE INSTRUCTION STREAM:

```

REPEAT = NO;
OPOVERRIDE = NO;
ADDROVERRIDE = NO;
POINTER = 0;

IF INSTR[POINTER] = F3H THEN          (*check for repeat prefix*)
    REPEAT = YES;
    POINTER = POINTER + 1;
ENDIF;
IF INSTR[POINTER] = 67H THEN          (*check for address size override
    ADDROVERRIDE = YES;                prefix*)
    POINTER = POINTER + 1;
ENDIF;
IF INSTR[POINTER] = 66H THEN          (*check for operand size override
    OPOVERRIDE = YES;                  prefix*)
    POINTER = POINTER + 1;
ENDIF;

IF INSTR[POINTER] IN( 2EH, 36H, 3EH, 26H, 64H, 65H) THEN (*check for OUTS
    OUTSEG = INSTR[POINTER];           segment overrides*)
    POINTER = POINTER + 1;
ELSE
    OUTSEG = 3EH;                      (*DS default segment*)
ENDIF;

OPCODE = INSTR[POINTER];              (*get the opcode*)
CASE (OPCODE) OF
    E4, EC : OPTYPE = IN; OPSIZE = BYTE;
    E5, ED : OPTYPE = IN; OPSIZE = WIDE;
    E6, EE : OPTYPE = OUT; OPSIZE = BYTE;
    E7, EF : OPTYPE = OUT; OPSIZE = WIDE;
    6C :      OPTYPE = INS; OPSIZE = BYTE
    6D :      OPTYPE = INS; OPSIZE = WIDE;
    6E :      OPTYPE = OUTS; OPSIZE = BYTE;
    6F :      OPTYPE = OUTS; OPSIZE = WIDE;
ENDCASE;

```

With the instruction opcode decoded, the operand and address sizes embedded in the instruction can now also be decoded. The instruction handlers will use the sizes to choose which registers to modify for proper re-execution (for example DI or EDI).

## DETERMINE OPERAND SIZE:

```

IF OPSIZE = WIDE THEN                (* determine word or dword *)
    IF OPOVERRIDE = YES THEN
        IF DEFOPERAND = 16 bits THEN
            OPSIZE = DWORD;
        ELSE
            OPSIZE = WORD;
        ENDIF;
    ELSE
        IF DEFOPERAND = 16 bits      (* no operand override *)
            OPSIZE = WORD;
        ELSE
            OPSIZE = DWORD;
        ENDIF;
    ENDIF;
ENDIF;

```

**DETERMINE ADDRESS SIZE:**

```

IF OPTYPE = INS OR OPTYPE = OUTS THEN
  IF ADDROVERRIDE = YES THEN
    IF DEFADDR = 16 bits THEN
      ADDRSIZE = 32 bits;
    ELSE
      ADDRSIZE = 16 bits;
    ENDIF;
  ELSE
    (* no override *)
    IF DEFADDR = 16 bits THEN
      ADDRSIZE = 16 bits;
    ELSE
      ADDRSIZE = 32 bits;
    ENDIF;
  ENDIF;
ENDIF;

```

To properly handle repeated I/O instructions, the state of the direction flag must be determined. The direction flag will determine how the instruction handlers set the index registers.

**GET DIRECTION FLAG:**

```

IF OPTYPE = INS OR OPTYPE = OUTS THEN
  IF DF = 0 (EFLG Bit 10, Loc 60004H) THEN
    DIRECTION = FORWARD;
  ELSE
    DIRECTION = BACKWARD;
  ENDIF;
ENDIF;

```

For use in restarting the REP OUTS instruction, the Overrun flag must be checked in the CPU save state. If this flag is clear, the REP OUTS instruction handler must account for the fact that two I/O cycles having taken place. If it is not, the handler calculates the index and count for only one I/O cycle.

**CHECK OVERRUN:**

```

IF OPTYPE = OUTS AND REPEAT = YES THEN
  IF OVERFLAG = 0 (Bit 0, Loc 6006CH) THEN
    OVERRUN = YES;
  ELSE
    OVERRUN = NO;
  ENDIF;
ENDIF;

```

With this information in hand, the SMM routine can now take any system dependent action required by the interrupt. When this action is complete, the SMM code is ready to resume the interrupted instruction stream. To accomplish this resumption, a separate course of action must be taken for each type of I/O instruction that was interrupted.

Depending on the type of I/O instruction trapped, a different instruction handler will be used to resume execution of the interrupted instruction stream. The following are pseudo-code handlers for each type of I/O instruction. Note that information created by the common front end code from the CPU save state and the instruction stream is used to determine how to properly prepare for I/O instruction resumption.

**IN, OUT Handler**—This instruction is the simplest to deal with because the instruction need only be re-executed. Therefore, the handler is reduced to assigning the save state EIP to the value of the last EIP stored in the CPU save state.

#### Opcodes IN:

E4 immed8 byte  
 E5 immed8 word/dword (depending on data size default or override)  
 EC byte (from [DX])  
 ED word/dword (from [DX] depending on data size)

#### Opcodes OUT:

E6 immed8 byte  
 E7 immed8 word/dword (depending on data size default or override)  
 EE byte (from [DX])  
 EF word/dword (from [DX] depending on data size)

#### Valid Overrides:

Opcode Size (66) on E5 and ED only to indicate size.

#### BEGIN IN

EIP = LAST EIP; (\*save state memory location changed\*)

#### END IN;

#### BEGIN OUT

EIP = LAST EIP; (\*save state memory location is changed\*)

#### END OUT;

**INS Handler**—The INS handler must account for operand and address sizes as well as the state of the direction flag. The EIP is set to the last EIP and either DI or EDI is set to the appropriate value depending on the address size.

#### Dependencies:

Address Size : 16 or 32 bit, depending on D bit and possible override  
 Operand Size : 8, 16, or 32 bit, depending on D bit and possible override for word form  
 Direction Flag : Affects String Increment or Decrement

#### Opcodes:

6C byte  
 6D word/dword (depending on data size)

#### Valid Overrides:

OpSize (66) on 6D form to indicate word or dword  
 AddrSize (67) to toggle default, select DI or EDI  
 Segment Overrides : Ignore/Invalid  
 REP—see REP INS case

#### BEGIN INS

EIP = LAST EIP; (\*save state memory location is changed\*)

#### CASE (OPSIZE) OF

BYTE : OFFSET = 1;  
 WORD : OFFSET = 2;  
 DWORD : OFFSET = 4;

#### ENDCASE;

#### IF DIRECTION = BACKWARD THEN

OFFSET = - OFFSET;

#### ENDIF;

#### IF ADDRSize = 16 BIT THEN

REG = DI;

#### ELSE

REG = EDI;

#### ENDIF;

REG = REG - OFFSET; (\* save state memory location is changed \*)

#### END INS;



**OUTS Handler**—The OUTS handler is similar to the INS handler and must also account for operand and address sizes as well as the state of the direction flag. The EIP is set to the Last EIP and either SI or ESI is set to the appropriate value depending on the address size. The “OUTSEG” variable holds the intended data segment for the transfer and does not need to be referenced in this example.

**Dependencies:**

Address Size : 16 or 32 bit, depending on D bit and possible override  
 Operand Size : 8, 16, or 32 bit, depending on D bit and possible override for word form  
 Direction Flag : Affects String Increment or Decrement

**Opcodes:**

6E byte  
 6F word/dword (depending on data size)

**Valid Overrides:**

OpSize (66) on 6F form to indicate word or dword  
 AddrSize (67) to toggle default, select SI or ESI  
 Segment Overrides : Any Segments allowed to override default DS for source  
 REP—see REP OUTS case

**BEGIN OUTS**

```
EIP = LAST EIP;          (*save state memory location is changed*)
CASE (OPSIZE) OF
  BYTE : OFFSET = 1;
  WORD : OFFSET = 2;
  DWORD : OFFSET = 4;
ENDCASE;
IF DIRECTION = BACKWARD THEN
  OFFSET = - OFFSET;
ENDIF;
```

**IF ADDRSize = 16 BIT THEN**

```
REG = SI;
ELSE
  REG = ESI;
ENDIF;
REG = REG-OFFSET;          (*save state memory location is changed*)
END OUTS;
```

**REP INS Handler**—The REP INS handler is similar to the INS handler because it must account for operand and address sizes as well as the state of the direction flag. The repeat count must also be calculated and set appropriately. The EIP is set to the Last EIP and either the pair DI/CX or EDI/ECX are set to the appropriate values depending on the address size.

**Dependencies:**

Address Size : 16 or 32 bit depending on D bit and possible override  
 Operand Size : 8, 16, or 32 bit depending on D bit and possible override for word form  
 Direction Flag : Affects String Increment or Decrement

**Opcodes:**

F3 6C byte  
 F3 6D word/dword (depending on data size)

**Valid Overrides:**

OpSize (66) on 6D form to indicate word or dword  
 AddrSize (67) to toggle default, select DI or EDI  
 Segment Overrides : Ignore/Invalid

```

BEGIN REPINS
    EIP = LAST EIP;                (*save state memory location is changed*)

    CASE (OPSIZE) OF
        BYTE : OFFSET = 1;
        WORD : OFFSET = 2;
        DWORD : OFFSET = 4 ;
    ENDCASE;

    IF DIRECTION = BACKWARD THEN
        OFFSET = - OFFSET;
    ENDIF;

    IF ADDRSIZE = 16 BIT THEN
        REG = DI;
        CNTREG = CX;
    ELSE
        REG = EDI;
        CNTREG = ECX;
    ENDIF;

    REG = REG - OFFSET;            (*save state memory location is changed*)
    CNTREG = CNTREG + 1;          (*save state memory location is changed*)
END REPINS;

```

**REP OUTS Handler**—The REP OUTS handler is the most complex since it must account for the possibility of an overrun condition in addition to operand and address sizes and the state of the direction flag. The EIP is set to the Last EIP and either the pair DI/CX or EDI/ECX are set to the appropriate values depending on the address sizes. The "OUTSEG" variable holds the intended data segment for the transfer and does not need to be referenced in this example.

**Dependencies:**

- Address Size : 16 or 32 bit depending on D bit and possible override
- Operand Size : 8, 16, or 32 bit depending on D bit and possible override for word form
- Direction Flag : Affects String Increment or Decrement
- Overrun Flag : Affects resetting of CX/ECX

**Opcodes:**

- F3 6E byte
- F3 6F word/dword (depending on data size)

**Valid Overrides:**

- OpSize (66) on 6D form to indicate word or dword
- AddrSize (67) to toggle default, select SI or ESI
- Segment Overrides : Any Segments allowed to override default DS for source

**BEGIN REPOUTS**

```

    EIP = LAST EIP;                (*save state memory location is changed*)

    CASE (OPSIZE) OF
        BYTE : OFFSET = 1;
        WORD : OFFSET = 2;
        DWORD : OFFSET = 4 ;
    ENDCASE;

    IF DIRECTION = BACKWARD THEN
        OFFSET = - OFFSET;
    ENDIF;

    IF ADDRSIZE = 16 BIT THEN
        REG = SI;
        CNTREG = CX;
    ELSE
        REG = ESI;
        CNTREG = ECX;
    ENDIF;

```

```

IF OVERRUN = NO THEN
    COUNT = 1;
ELSE
    OFFSET = 2 * OFFSET;
    COUNT = 2;
ENDIF;

REG = REG - OFFSET;      (*save state memory location is changed*)
CNTREG = CNTREG + COUNT; (*save state memory location is changed*)
END REPOUTS;

```

### SMM STATUS BIT

Before exiting SMM, the SMM code must be sure to clear the System Management Mode Status bit (SMMS) in the debug status register (DR6, bit 12). Some diagnostic software checks this register and may indicate a CPU failure if the SMMS bit is not cleared before return. The definition of the lower 16 bits of the Debug Status Register is given in Table 2-7 for reference.

**Table 2-7 Debug Register 6 (Low word only)**

|    |    |    |      |    |    |   |   |   |   |   |   |    |    |    |    |
|----|----|----|------|----|----|---|---|---|---|---|---|----|----|----|----|
| 15 | 14 | 13 | 12   | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3  | 2  | 1  | d0 |
| BT | BS | BD | SMMS | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | B3 | B2 | B1 | B0 |

SMMS = 1 SMM was entered  
 SMMS = 0 SMM status cleared

It should be noted that nothing prevents software from setting the SMMS bit. This can cause confusion in determining the source of the status change and should not be done. The protocol of allowing the CPU to set the bit and clearing the status with the SMM code should be maintained.

### HANDLING MULTIPLE EVENTS

At the completion of the appropriate instruction handler, the SMM code may include a section to check for pending non-trap events and handle these events before exiting SMM. If there are no other pending SMI requests then the SMM code need only issue the resume command to exit SMM and continue proper execution of the interrupted instruction stream.

### SOFTWARE SMI GENERATION

Besides hardware initiation of the system management interrupt via the  $\overline{\text{SMI}}$  pin, there is also a software induced SMI mechanism. Generating a soft SMI involves setting control bit SMIE in Debug Control Register (DR7, bit 12) and executing a reserved opcode. The definition of the lower 16 bits of the Debug Control Register is given in Table 2-8 for reference.

A logic one written to the SMIE control bit enables the soft SMI opcode. The SMIE bit is located at bit 12 in DR7. The default (reset) state of SMIE is zero. The other reserved bits in DR7 remain as previously defined.

The soft SMI opcode is 0F1h. If the SMIE bit is a one, execution of opcode 0F1h generates a soft SMI. If the SMIE bit is a zero, then execution of this opcode generates a standard Interrupt 1 (Debug Exception). The code fragment in Figure 2-11 shows an example of how to generate a soft SMI.

After execution of the SMM routine, normal code execution resumes at the instruction following the 0F1h opcode. The SMIE bit should be set back to zero after the execution of the 0F1h opcode so that any errant execution of an 0F1h opcode by application software behaves the same as in a non-SMI based system.

The functional sequence of the software-based SMI is identical to the hardware-based SMI with the exception that the SMI pin is not initially driven active by an external source. Upon execution of a soft SMI opcode, the SMI pin is driven active (Low) by the processor before the save state operation begins. It should be noted for Protected Mode use that the soft SMI opcode is not a privileged instruction.

**Figure 2-11 Soft SMI Generation**

```
MOV EAX,DR7      ;get current DR7 contents
OR  EAX,00001000h ;set SMIE bit (#12)
MOV DR7,EAX      ;load DR7 with SMIE set
DB  0F1h         ;execute soft SMI
```

**Table 2-8 Debug Register 7 (Low word only)**

|    |    |    |      |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|------|----|----|----|----|----|----|----|----|----|----|----|----|
| 15 | 14 | 13 | 12   | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |
| 0  | 0  | 0  | SMIE | 0  | 0  | GE | LE | G3 | L3 | G2 | L2 | G1 | L1 | G0 | L0 |

SMIE = 1 enables soft SMI  
SMIE = 0 disables soft SMI

## HARDWARE ISSUES

The Am386DXLV and Am386SXLV microprocessors provide an unprecedented opportunity for product differentiation due to the general nature of the SMM solution. System logic hardware support for SMM comprehends both basic SMM interface support as well as I/O trapping via the I/O Instruction Break Enable feature. The following design guidelines and examples are provided as a basis for planning new SMM support logic.

## SMM Signal Design Guidelines

|                      |   |
|----------------------|---|
| <b><u>SMI</u></b>    | This signal is a synchronous input, which does not have the multiple stages of synchronization of the NMI and INTR interrupt inputs. <u>SMI</u> is also bidirectional and is pulled up by a weak internal pull-up when not driven by the CPU. This pull-up disabled when the CPU uses <u>SMI</u> as an output to conserve power. After the processor drives the SMI signal High (inactive), external logic should allow <u>SMI</u> to be High two CLK2 periods before driving it Low again. The CPU drives <u>SMI</u> High during these CLK2 periods, thereby eliminating the larger current demand usually associated with open collector protocols. |
| <b><u>SMIADS</u></b> | This pin is a standard CPU output. No special treatment is required. Note that the signal floats in response to a HOLD state.   |
| <b><u>SMIRDY</u></b> | This pin is a standard synchronous CPU input with an internal pull-up. Despite its similar function to READY, the two signals cannot be sourced from the same signal. System logic should terminate <u>SMIADS</u> initiated accesses by asserting <u>SMIRDY</u> . READY is ignored during <u>SMIADS</u> initiated cycles. Terminating ADS initiated cycles with <u>SMIRDY</u> results in unpredictable CPU behavior.  |
| <b><u>IIBEN</u></b>  | This signal is an asynchronous CPU input with active pull-up. <u>IIBEN</u> is synchronized by CLK2 internal to the CPU. The <u>IIBEN</u> pull-up is active during RESET pulses and whenever the signal is not driven active by the system. To conserve power, the pull-up is disabled by the CPU whenever the CPU detects that the system logic is driving <u>IIBEN</u> Low.  |

### CHIPSET INDEPENDENT DESIGN EXAMPLE

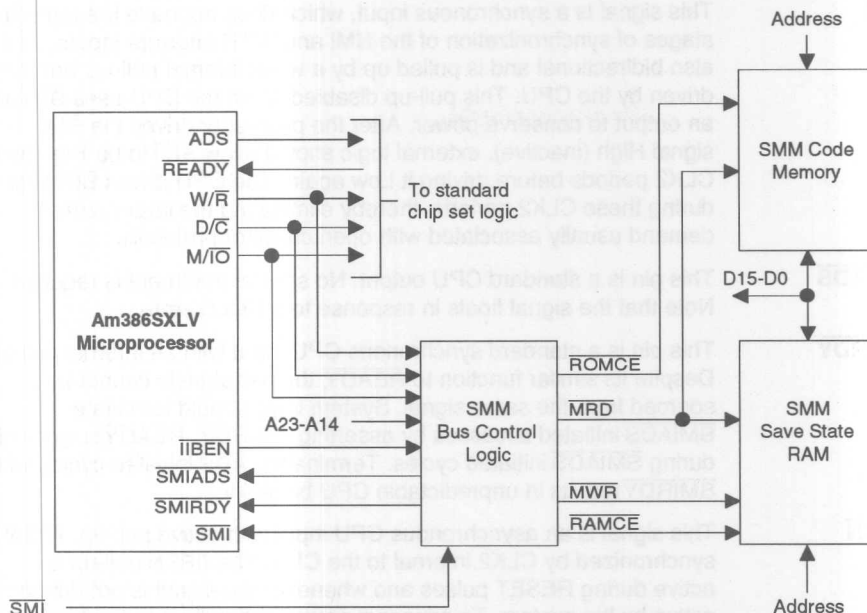
Figure 2-12 shows an Am386SXLV process-based system implementation that is independent of the normal system control logic (a chip-set independent design).

This design can be implemented on a CPU daughter-card or on a motherboard with existing chip sets. The design example consists of an Am386SXLV CPU, SMM Bus Control Logic, an SMM Code ROM, and an SMM State Save RAM.

The SMM Bus Control Logic contains a bus control state machine that starts cycles upon receiving an SMIADS, generates command pulses with the appropriate number of wait states for the SMM RAM or ROM, and then generates an SMIRDY pulse. The logic also contains the address decoder for the ROM and RAM chip enables. The ROM chip enable should be active for a predefined code space, along with an alias in the last 16 bytes of the ROM for the initial FFFFF0h code fetches. The RAM chip enable should be active for at least the physical address range of 60000h–601FFh for the CPU save state table.

The SMM Code ROM is a simple 16-bit ROM implementation, with all bytes always driven on the D15–D0 pins on all ROM reads. The SMI State Save RAM is a simple 16-bit RAM implementation, with all bytes always written or read during bus transfer cycles to or from the RAM. Simple 16-bit capability is all that is required for the CPU state save and restore. If the RAM is also to be used as a “scratch pad” memory, then individual byte write cycles can be supported by factoring in the byte enable lines (BHE and BLE) into separate write enables per byte. An Am386DXLV processor-based design would have a 32-bit ROM and RAM subsystem. All bus control logic would be the same.

**Figure 2-12 SMM Function Discrete Implementation**



16944A-13

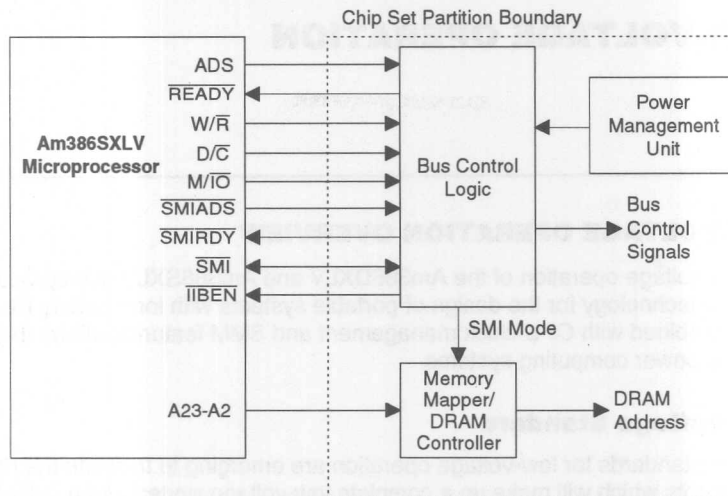
The bus control logic should also include a register for the CPU to identify the SMI source. The SMM routine can use this information to determine its course of action dependent on whether the source was an I/O trap event, soft SMI, or general SMI.

### CHIP SET INTEGRATION

Figure 2-13 shows a suggested Am386SXLV processor-based system implementation that could be included in the design of the normal system control logic (SMM integration within the chip set). The two main areas in the system control logic that would be affected are the bus control logic and the memory mapper of the DRAM controller.

The Bus Control Logic would have additional connections for the SMIADS, SMIRDY, SMI, and IIBEN pins. It should generate the standard control signals for SMM bus cycles while indicating to the DRAM controller that SMM address mapping should be done for the cycle. The bus controller could also convert the 16-bit only cycles to 8-bit conversion cycles in order to support a smaller bus width for the SMI Code ROM and Save State RAM if they are separate from the system DRAM.

The DRAM controller can be designed to map SMM addresses into a protected area of system DRAM. A good example area would be unused Shadow RAM space. Both the SMM code and Save State addresses could be mapped into a single 16-KB to 128-KB segment of system DRAM. With the exception of the save state and SMM code memory area, the rest of the SMM address space can be mapped one for one onto the normal address space. This allows the use of standard memory reference instructions to access data areas and BIOS routines.

**Figure 2-13 SMM System Logic Functions**

16944A-14

The system power management unit should route its system management related interrupts to the Bus Control Logic. This solution allows the PMU to be asynchronous to the system clock. The bus control logic is then responsible for generating the synchronous SMI signal.

The Bus Control Logic should also include a register for the CPU to identify the SMI source. The SMM routine uses this information to determine its course of action dependent on whether the source was an I/O trap event, soft SMI, or general SMI.

#### SOFT SMI ALTERNATIVES

In cases where it is desirable to implement software initiated SMI routines, the system designer can use the soft SMI request opcode just as the software requested INT N command functions for normal interrupts. However, it may be desirable to implement a register in the system logic which asserts the SMI signal in response to an I/O register write.

The system core logic designer should include a register that indicates how the interrupt was generated (whether the source was an I/O trap or not) and what system event needs service if the event was not an I/O trap.





## LOW-VOLTAGE OPERATION

### LOW-VOLTAGE OPERATION OVERVIEW

The low-voltage operation of the Am386DXLV and Am386SXLV microprocessors is an enabling technology for the design of portable systems with long battery life. This capability, combined with CPU clock management and SMM features, allows the design of very low power computing systems.

### Low-Voltage Standard

Industry standards for low-voltage operation are emerging to facilitate the design of components which will make up a complete low-voltage system. As a guideline, the Am386DXLV and Am386SXLV processor specifications follow the first article or regulated version of the JEDEC 8.0 low-voltage proposal. This standard proposal calls for a  $V_{cc}$  range of  $3.3 \text{ V} \pm 10\%$ . To ease the design of a mixed voltage system, the standard also supports CMOS and TTL outputs.

### Power Savings

CMOS Dynamic power consumption is proportional to the square of the operating voltage multiplied by capacitance and operating frequency. Static CPU operation can reduce power consumption by enabling the system designer to reduce operating frequency when possible. However, operating voltage is always the dominant factor in power consumption. By reducing the operating voltage from 5 V to 3.3 V for any device, the power consumed is reduced by 56% (see Figure 3-1).

**Figure 3-1 3-V and 5-V System Dynamic Power Consumption**

#### 3.3 V

$$\begin{aligned} P_3 &= V^2CFK \\ P_3 &= K(3.3)^2CFK \\ P_3 &= 10.89CFK \end{aligned}$$

#### 5 V

$$\begin{aligned} P_5 &= V^2CFK \\ P_5 &= (5)^2CFK \\ P_5 &= 25CFK \end{aligned}$$

$$\begin{aligned} \text{Reduction} &= 1 - P_3/P_5 \\ &= 1 - 0.44 \\ &= 56\% \end{aligned}$$

where V = Voltage  
C = Capacitance  
F = Frequency  
K = Constant

The reduction of CPU and core logic operating voltage dramatically reduces overall system power consumption. Additional power savings can be realized as low-voltage mass storage and peripheral devices become available.

Two basic strategies exist in designing systems containing the Am386DXLV and Am386SXLV microprocessors. The first strategy is to design a homogenous system in which all logic components operate at 3.3 V. This provides the best overall power consumption. However, system designers may need to include devices for which 3.3-V versions are not available. In the second strategy, the system designer must then design a mixed 5-V/3.3-V system. This compromise allows the system designer to minimize the core logic power consumption while still including the functionality of the 5-V features. The choice of a mixed voltage system design also involves balancing design complexity with the need for the additional features.

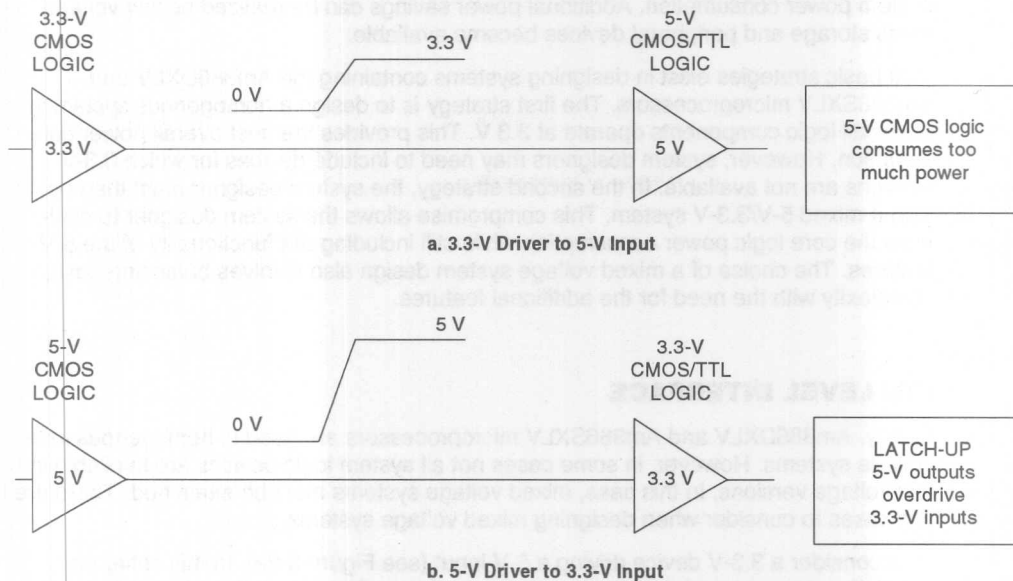
## PIN LEVEL INTERFACE

Ideally, Am386DXLV and Am386SXLV microprocessors are used in homogenous low-voltage systems. However, in some cases not all system logic devices are available in low-voltage versions. In this case, mixed voltage systems must be examined. There are two cases to consider when designing mixed voltage systems.

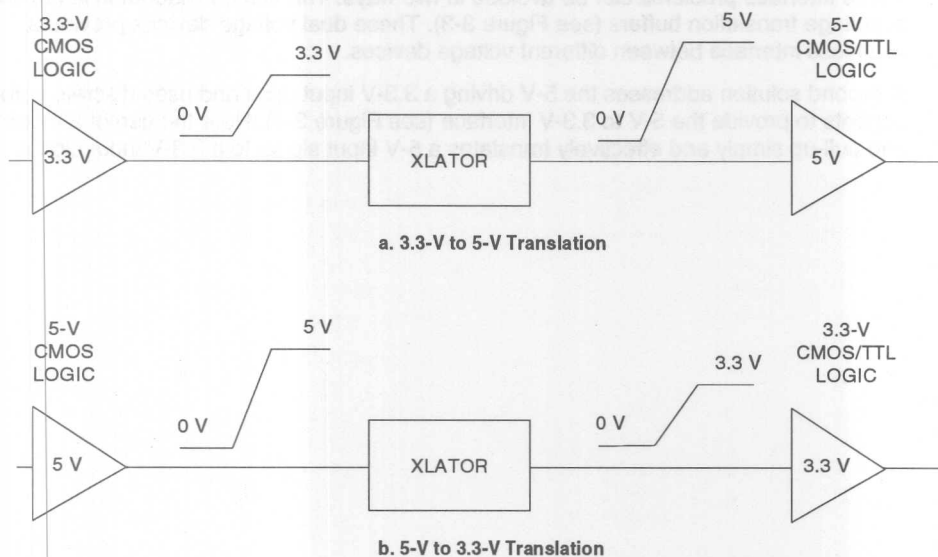
First consider a 3.3-V device driving a 5-V input (see Figure 3-2a). In this case, the 3.3-V signal is subject to lower noise immunity than a 5-V signal. If the buffer has a pure CMOS input, the 3.3-V signal does not drive the input buffer completely out of the transition region, thereby allowing excessive current to be consumed. The second case, a 5-V device driving a 3.3-V input (see Figure 3-2b) poses a more serious problem. In this case, the 5-V signal will over drive and possibly breakdown the 3.3-V input. This breakdown can lead to potentially damaging latch up of the 3.3-V device.

These interface problems can be avoided in two ways. The first consideration is the use of voltage translation buffers (see Figure 3-3). These dual voltage devices provide a seamless interface between different voltage devices.

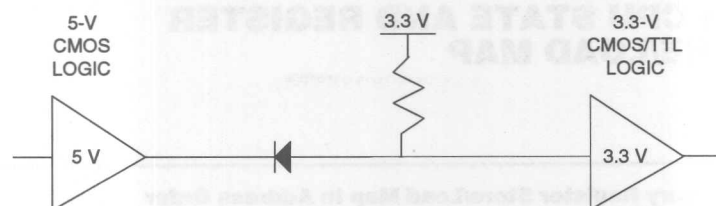
A second solution addresses the 5-V driving a 3.3-V input case and uses discrete components to provide the 5-V to 3.3-V interface (see Figure 3-4). Here the use of a diode and pull-up simply and effectively translates a 5-V input signal to a 3.3-V input signal.

**Figure 3-2 Mixed 5-V/3-V Considerations**


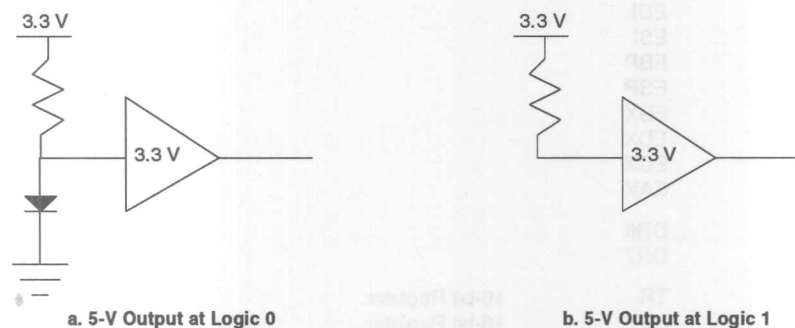
16944A-16

**Figure 3-3 Mixed System with Voltage Translators**


16944A-17


**Figure 3-4 Mixed System with Discrete Translators**


16944A-18

**Figure 3-5 Discrete Translator Function**


16944A-019

The operation of this discrete translator is diagramed in Figure 3-5. When the 5-V signal is Low, the diode is forward biased and the pull-up current is shunted through the 5-V driver (see Figure 3-5a). Thus, a low voltage is presented to the 3.3-V input. When the 5-V signal is High, the diode is reverse biased with respect to the 3.3-V pull up and is thus blocked from the 3.3-V input (see Figure 3-5b); therefore, the 3.3-V pull-up is presented to the input as a valid High. The choice of the pull-up value must balance the AC timing requirements of the signal with the desire for low current consumption.

Optionally, the use of discrete components can be eliminated by understanding the type of devices needed to interface to 3.3-V and 5-V logic and including the translation into the system core logic. Signals requiring a 5-V interface can be driven by circuitry with 5-V supplies. Any 3.3-V devices can be driven by circuitry with 3.3-V supplies. In this solution the core logic isolates different parts of the interface logic to drive the appropriate levels.



## SMM CPU STATE AND REGISTER STORE/LOAD MAP

**Table A-1 SMI Memory Register Store/Load Map In Address Order**

| SMM Address | Registers | Comments              |
|-------------|-----------|-----------------------|
| 60000       | CR0       |                       |
| 60004       | EFLAGS    |                       |
| 60008       | EIP       |                       |
| 6000C       | EDI       |                       |
| 60010       | ESI       |                       |
| 60014       | EBP       |                       |
| 60018       | ESP       |                       |
| 6001C       | EBX       |                       |
| 60020       | EDX       |                       |
| 60024       | ECX       |                       |
| 60028       | EAX       |                       |
| 6002C       | DR6       |                       |
| 60030       | DR7       |                       |
| 60034       | TR        | 16-bit Register       |
| 60038       | LDT       | 16-bit Register       |
| 6003C       | GS        | 16-bit Register       |
| 60040       | FS        | 16-bit Register       |
| 60044       | DS        | 16-bit Register       |
| 60048       | SS        | 16-bit Register       |
| 6004C       | CS        | 16-bit Register       |
| 60050       | ES        | 16-bit Register       |
| 60054       | SA:TSS    | ATTR. TASK DESCR      |
| 60058       | SB:TSS    | BASE ADDR TSS         |
| 6005C       | SL:TSS    | ADDR LIMIT TSS        |
| 60060       |           | Reserved              |
| 60064       | SB:IDT    | BASE ADDR IDT         |
| 60068       | SL:IDT    | ADDR LIMIT IDT        |
| 6006C       | IOFLAG    | REP OUTS Overrun Flag |
| 60070       | SB:GDT    | BASE ADDR             |
| 60074       | SL:GDT    | ADDR LIMIT            |

**Table A-1 SMI Memory Register Store/Load Map In Address Order (continued)**

| SMM Address | Registers | Comments           |
|-------------|-----------|--------------------|
| 60078       | SA:LDT    | ATTR LDT DESCR.    |
| 6007C       | SB:LDT    | BASE ADDR LDT      |
| 60080       | SL:LDT    | ADDR LIMIT LDT     |
| 60084       | SA:GS     | ATTR GS DESCR      |
| 60088       | SB:GS     | BASE ADDR GS       |
| 6008C       | SL:GS     | ADDR LIMIT GS      |
| 60090       | SA:FS     | ATTR FS DESCR      |
| 60094       | SB:FS     | BASE ADDR FS       |
| 60098       | SL:FS     | ADDR LIMIT FS      |
| 6009C       | SA:DS     | ATTR DS DESCR      |
| 600A0       | SB:DS     | BASE ADDR DS       |
| 600A4       | SL:DS     | ADDR LIMIT DS      |
| 600A8       | SA:SS     | ATTR SS DESCR      |
| 600AC       | SB:SS     | BASE ADDR SS       |
| 600B0       | SL:SS     | ADDR LIMIT SS      |
| 600B4       | SA:CS     | ATTR CS DESCR      |
| 600B8       | SB:CS     | BASE ADDR          |
| 600BC       | SL:CS     | ADDR LIMIT         |
| 600C0       | SA:ES     | ATTR ES DESCR      |
| 600C4       | SB:ES     | BASE ADDR          |
| 600C8       | SL:ES     | ADDR LIMIT         |
| 60100       |           | Temporary Register |
| 60104       |           | Temporary Register |
| 60108       |           | Temporary Register |
| 6010C       |           | Temporary Register |
| 60110       |           | Temporary Register |
| 60114       |           | Temporary Register |
| 60118       |           | Temporary Register |
| 6011C       |           | Temporary Register |
| 60120       |           | Temporary Register |
| 60124       | LEIP      | Last EIP           |

- Notes:
1. For the overrun flag at 6006CH, all bits 15–1 are reserved by AMD for future use.
  2. All bits of 60060 are reserved by AMD.
  3. Unused bits in 16-bit writes reserved by AMD.
  4. Definitions:
    - SA—Segment Attribute
    - SB—Segment Base
    - SL—Segment Limits

**Table A-2 SMI Memory Register Store/Load Map In Bus Cycle Order**

| SMM Address | Registers | Comments              |
|-------------|-----------|-----------------------|
| 60000       | CR0       |                       |
| 60100       |           | Temporary Register    |
| 60104       |           | Temporary Register    |
| 60108       |           | Temporary Register    |
| 6010C       |           | Temporary Register    |
| 60110       |           | Temporary Register    |
| 60114       |           | Temporary Register    |
| 60118       |           | Temporary Register    |
| 6011C       |           | Temporary Register    |
| 60120       |           | Temporary Register    |
| 60124       | LEIP      | Last EIP              |
| 60004       | EFLAGS    |                       |
| 60008       | EIP       |                       |
| 6000C       | EDI       |                       |
| 60010       | ESI       |                       |
| 60014       | EBP       |                       |
| 60018       | ESP       |                       |
| 6001C       | EBX       |                       |
| 60020       | EDX       |                       |
| 60024       | ECX       |                       |
| 60028       | EAX       |                       |
| 6002C       | DR6       |                       |
| 60030       | DR7       |                       |
| 60034       | TR        | 16-bit Register       |
| 60038       | LDT       | 16-bit Register       |
| 6003C       | GS        | 16-bit Register       |
| 60040       | FS        | 16-bit Register       |
| 60044       | DS        | 16-bit Register       |
| 60048       | SS        | 16-bit Register       |
| 6004C       | CS        | 16-bit Register       |
| 60050       | ES        | 16-bit Register       |
| 60054       | SA:TSS    | ATTR TASK DESCR       |
| 60058       | SB:TSS    | BASE ADDR TSS         |
| 6005C       | SL:TSS    | ADDR LIMIT TSS        |
| 60060       |           | Reserved              |
| 60064       | SB:IDT    | BASE ADDR IDT         |
| 60068       | SL:IDT    | ADDR LIMIT IDT        |
| 6006C       | IOFLAG    | REP OUTS Overrun Flag |
| 60070       | SB:GDT    | BASE ADDR             |
| 60074       | SL:GDT    | ADDR LIMIT            |
| 60078       | SA:LDT    | ATTR LDT DESCR        |
| 6007C       | SB:LDT    | BASE ADDR LDT         |
| 60080       | SL:LDT    | ADDR LIMIT LDT        |



**Table A-2 SMI Memory Register Store/Load Map In Bus Cycle Order (continued)**

| SMM Address | Registers | Comments      |
|-------------|-----------|---------------|
| 60084       | SA:GS     | ATTR GS DESCR |
| 60088       | SB:GS     | BASE ADDR GS  |
| 6008C       | SL:GS     | ADDR LIMIT GS |
| 60090       | SA:FS     | ATTR FS DESCR |
| 60094       | SB:FS     | BASE ADDR FS  |
| 60098       | SL:FS     | ADDR LIMIT FS |
| 6009C       | SA:DS     | ATTR DS DESCR |
| 600A0       | SB:DS     | BASE ADDR DS  |
| 600A4       | SL:DS     | ADDR LIMIT DS |
| 600A8       | SA:SS     | ATTR SS DESCR |
| 600AC       | SB:SS     | BASE ADDR SS  |
| 600B0       | SL:SS     | ADDR LIMIT SS |
| 600B4       | SA:CS     | ATTR CS DESCR |
| 600B8       | SB:CS     | BASE ADDR     |
| 600BC       | SL:CS     | ADDR LIMIT    |
| 600C0       | SA:ES     | ATTR ES DESCR |
| 600C4       | SB:ES     | BASE ADDR     |
| 600C8       | SL:ES     | ADDR LIMIT    |

- Notes:
1. For the overrun flag at 6006CH, all bits 15-1 are reserved by AMD for future use.
  2. All bits of 60060 are reserved by AMD.
  3. Unused bits in 16-bit writes reserved by AMD.
  4. Definitions: SA-Segment Attribute  
SB-Segment Base  
SL-Segment Limits

**Table A-3 Processor State Upon Entering SMM**
**Value of Saved Register**

| Register | Value   |
|----------|---|
| CR0      | 0   |
| EFLAGS   | 00000002  |
| EIP      | 0000FFF0  |
| EDI      | 0   |
| ESI      | 0   |
| EBP      | 0   |
| ESP      | 0   |
| EBX      | 0   |
| EDX      | 0   |
| ECX      | 0   |
| EAX      | 0   |
| DR6      | Previous DR6 w/bit 12 = 1 and bits 3–0 reflect debug reg 0–3 status |
| DR7      | 0   |
| TR       | 0   |
| LDT      | 0   |
| GS       | 0   |
| FS       | 0   |
| DS       | 0   |
| SS       | 0   |
| CS       | F000  |
| ES       | 0   |
| CPL      | 0   |

Note: Page Unit TLB – FLUSHED

**Table A-3 Processor State Upon Entering SMM (continued)****Values of Internal Descriptor Cache**

| Register | Value       |
|----------|-------------|
| SA:TSS   | 00008200    |
| SB:TSS   | 0           |
| SL:TSS   | FFFFFFFF    |
| SA:IDT   | none exists |
| SB:IDT   | 0           |
| SL:IDT   | 0000FFFF    |
| SA:GDT   | none exists |
| SB:GDT   | 0           |
| SL:GDT   | 0000FFFF    |
| SA:GS    | 00008200    |
| SB:GS    | 0           |
| SL:GS    | 0000FFFF    |
| SA:FS    | 00008200    |
| SB:FS    | 0           |
| SL:FS    | 0000FFFF    |
| SA:DS    | 00008200    |
| SB:DS    | 0           |
| SL:DS    | 0000FFFF    |
| SA:SS    | 00008200    |
| SB:SS    | 0           |
| SL:SS    | 0000FFFF    |
| SA:CS    | 00008200    |
| SB:CS    | FFFF0000    |
| SL:CS    | 0000FFFF    |
| SA:ES    | 00008200    |
| SB:ES    | 0           |
| SL:ES    | 0000FFFF    |

Note: 1. Definitions: SA—Segment Attribute  
SB—Segment Base  
SL—Segment Limits

Page Unit TLB – FLUSHED

**Table A-3 Processor State Upon Entering SMM (continued)**
**Values of Non-Saved Registers**

| Register | Value     |
|----------|-----------|
| DR0      | unchanged |
| DR1      | unchanged |
| DR2      | unchanged |
| DR3      | unchanged |
| CR2      | unchanged |
| CR3      | unchanged |
| TR6      | unchanged |
| TR7      | unchanged |



## UMOV ASSEMBLER MACROS

The following listing can be used to construct the various forms of the UMOV instruction.

```

;*****MACROS*****
; UMOV Code Builder Macros
;*****
;
;  MACRO INVOCATION/FORMATS:
;
;      umovrdw preg,pmem,padder ;UMOV memory read, word/doubleword
;      umovrd8 preg,pmem,padder ;UMOV memory read, byte, real mode
;      umovrd8p preg,pmem,padder ;UMOV memory read, 4.25byte,32-bit prot
;      umovwrw pmem,preg,padder ;UMOV memory write,word/doubleword
;      umovwr8 pmem,preg,padder ;UMOV memory write,byte, real mode
;      umovwr8p pmem,preg,padder ;UMOV memory write,byte, 32-bit prot
;
;
;  PARAMETERS:
;
;      preg - register specification, (i.e. AX,AL,EAX,SI,etc.)
;
;      pmem - memory address spec, one of the 386 addressing modes,
;              example would be [SI] or [EBX][ECX*4]+10
;
;      padder - displacement spec: this field specifies the added length
;              of the instruction due to the addressing mode, (i.e., the
;              number of bytes to add to the instruction besides the
;              basic ModRegR/M. Displacements and the SIB byte add to
;              the length).
;              This field should be one of the following:
;
;
;      Keyword  Adder  Example
;      -----
;      nod      0      umovrdw ax,[si],nod
;      d8       1      umovrd8 bl,[di+055h],d8
;      d16      2      umovwrw [bx+0AAAAh],dx,d16
;      d32      4      umovwr8 [ebx+0100000h],cx,d32
;      sib      1      umovrdw eax,[ebx][esi*2]
;      sibd8    2      umovwrw [ebx][ecx*4]+5,eax,sibd8
;      sibd32   5      umovwr8 [ebx][ecx*4]+100000H,dh,sibd32
;
;
;  WARNINGS:
;      For the word and doubleword versions of the macros, all address
;      and operand sizing prefixes will be properly generated.
;      However, with the byte-sized versions, care must
;      be taken to ensure that incorrect address size overrides don't
;      get generated. To solve this problem, two versions of the byte
;      size macros are supplied. UMOV8 and UMOV8P are for use in
;      16-bit code segments (those with the USE16 segment attribute).
;      UMOV8D and UMOV8DP are for use in 32-bit Protected-Mode code
;      segments (those with the USE32 segment attribute).

```

```

; COMMENTS:
;
; - Currently, only lowercase keywords are recognized for the
;   padder field.
;
; - Note that segment register overrides ARE ALLOWED in these
;   macros:
;
;               umovrd8 al,es:[di],nod
;
;*****
;*
;* UMOV READ WIDE (16/32) FROM MEMORY TO REGISTER
;*
;*****
umovrdw macro preg,pmem,padder
    local istart,inext,dispcnt,iprefix,ysize,iopcode

;;figure out displacement field size
    IFIDN <padder>,<nod>
        dispcnt = 0
    ELSE
        IFIDN <padder>,<d8>
            dispcnt = 1
        ELSE
            IFIDN <padder>,<d16>
                dispcnt = 2
            ELSE
                IFIDN <padder>,<d32>
                    dispcnt = 4
                ELSE
                    IFIDN <padder>,<sib>
                        dispcnt = 1
                    ELSE
                        IFIDN <padder>,<sibd8>
                            dispcnt = 2
                        ELSE
                            IFIDN <padder>,<sibd32>
                                dispcnt = 5
                            ELSE
                                .ERR
                            ENDIF
                        ENDIF
                    ENDIF
                ENDIF
            ENDIF
        ENDIF
    ENDIF

;;start code generation
    istart = $
    lar     preg,pmem
    inext   = $
    ysize   = $-istart
    iprefix = ysize - dispcnt - 3
    iopcode = istart + iprefix + 1
    org iopcode
    db      13H                ;;locate for overlay opcode
                                ;;umove opcode
    org inext                ;;go back to next
    endm

```



```

;*****
;*
;* UMOV READ BYTE (8 BIT) FROM MEMORY TO REGISTER, 16-BIT CS
;*
;*****
umovrd8 macro preg,pmem,padder
    local istart,inext,dispcnt,iprefix,ysize,iopcode,sreg

;;figure out displacement field size
    IFIDN <padder>,<nod>
        dispcnt = 0
    ELSE
        IFIDN <padder>,<d8>
            dispcnt = 1
        ELSE
            IFIDN <padder>,<d16>
                dispcnt = 2
            ELSE
                IFIDN <padder>,<d32>
                    dispcnt = 4
                ELSE
                    IFIDN <padder>,<sib>
                        dispcnt = 1
                    ELSE
                        IFIDN <padder>,<sibd8>
                            dispcnt = 2
                        ELSE
                            IFIDN <padder>,<sibd32>
                                dispcnt = 5
                            ELSE
                                .ERR
                            ENDIF
                        ENDIF
                    ENDIF
                ENDIF
            ENDIF
        ENDIF
    ENDIF

;;figure out register substitution
    IFIDN <preg>,<al>
        sreg EQU <AX>
    ELSE
        IFIDN <preg>,<bl>
            sreg EQU <BX>
        ELSE
            IFIDN <preg>,<cl>
                sreg EQU <CX>
            ELSE
                IFIDN <preg>,<dl>
                    sreg EQU <DX>
                ELSE
                    IFIDN <preg>,<ah>
                        sreg EQU <SP>
                    ELSE
                        IFIDN <preg>,<bh>
                            sreg EQU <DI>
                        ELSE
                            IFIDN <preg>,<ch>
                                sreg EQU <BP>
                            ELSE
                                IFIDN <preg>,<dh>
                                    sreg EQU <SI>
                                ELSE
                                    .ERR
                                ENDIF
                            ENDIF
                        ENDIF
                    ENDIF
                ENDIF
            ENDIF
        ENDIF
    ENDIF

```



```

        ENDIF
        ENDIF
;;start code generation
istart = $
        lar     sreg,pmem
inext   = $
isize   = $-istart
iprefix = isize - dispcnt - 3
iopcode = istart + iprefix + 1
        org iopcode      ;;locate for overlay opcode
        db    12H        ;;umove opcode
        org inext        ;;go back to next
        endm

;*****
;*
;* UMOV READ BYTE (8 BIT) FROM MEMORY TO REGISTER, 32-BIT CS
;*
;*****

umovrd8p macro preg,pmem,padder
        local istart,inext,dispcnt,iprefix,isize,iopcode,sreg
;;figure out displacement field size
        IFIDN <padder>,<nod>
dispcnt = 0
        ELSE
        IFIDN <padder>,<d8>
dispcnt = 1
        ELSE
        IFIDN <padder>,<d16>
dispcnt = 2
        ELSE
        IFIDN <padder>,<d32>
dispcnt = 4
        ELSE
        IFIDN <padder>,<sib>
dispcnt = 1
        ELSE
        IFIDN <padder>,<sibd8>
dispcnt = 2
        ELSE
        IFIDN <padder>,<sibd32>
dispcnt = 5
        ELSE
        .ERR
        ENDIF
        ENDIF
        ENDIF
        ENDIF
        ENDIF
        ENDIF
        ENDIF
        ENDIF
;;figure out register substitution
        IFIDN <preg>,<al>
sreg EQU <EAX>
        ELSE
        IFIDN <preg>,<bl>
sreg EQU <EBX>
        ELSE
        IFIDN <preg>,<cl>
sreg EQU <ECX>
        ELSE
        IFIDN <preg>,<dl>
sreg EQU <EDX>
        ELSE
        IFIDN <preg>,<ah>
sreg EQU <ESP>
        ELSE
        IFIDN <preg>,<bh>
sreg EQU <EDI>

```

```

ELSE
IFIDN <preg>,<ch>
sreg EQU <EBP>
ELSE
IFIDN <preg>,<dh>
sreg EQU <ESI>
ELSE
.ERR
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
;;start code generation
istart = $
lar sreg,pmem
inext = $
isize = $-istart
iprefix = isize - dispcnt - 3
iopcode = istart + iprefix + 1
org iopcode ;;locate for overlay opcode
db 12H ;;umove opcode
org inext ;;go back to next
endm

;*****
;*
;* UMOV WRITE WIDE (16/32) TO MEMORY FROM REGISTER
;*
;*****
umovwrw macro pmem, preg, padder
local istart, inext, dispcnt, iprefix, isize, iopcode
;;figure out displacement field size
IFIDN <padder>,<nod>
dispcnt = 0
ELSE
IFIDN <padder>,<d8>
dispcnt = 1
ELSE
IFIDN <padder>,<d16>
dispcnt = 2
ELSE
IFIDN <padder>,<d32>
dispcnt = 4
ELSE
IFIDN <padder>,<sib>
dispcnt = 1
ELSE
IFIDN <padder>,<sibd8>
dispcnt = 2
ELSE
IFIDN <padder>,<sibd32>
dispcnt = 5
ELSE
.ERR
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF

```

```

;;start code generation
istart = $
        lar      preg,pmem
inext  = $
isize  = $-istart
iprefix = isize - dispcnt - 3
iopcode = istart + iprefix + 1
        org iopcode      ;;locate for overlay opcode
        db      11H      ;;umove opcode
        org inext        ;;go back to next
        endm

;*****
;*
;* UMOV WRITE BYTE (8 BIT) TO MEMORY FROM REGISTER, 16-BIT CS
;*
;*****
umovwr8 macro pmem,preg,padder
        local istart,inext,dispcnt,iprefix,isize,iopcode,sreg
        ;;figure out displacement field size
        IFIDN <padder>,<nod>
        dispcnt = 0
        ELSE
        IFIDN <padder>,<d8>
        dispcnt = 1
        ELSE
        IFIDN <padder>,<d16>
        dispcnt = 2
        ELSE
        IFIDN <padder>,<d32>
        dispcnt = 4
        ELSE
        IFIDN <padder>,<sib>
        dispcnt = 1
        ELSE
        IFIDN <padder>,<sibd8>
        dispcnt = 2
        ELSE
        IFIDN <padder>,<sibd32>
        dispcnt = 5
        ELSE
        .ERR
        ENDIF
        ENDIF
        ENDIF
        ENDIF
        ENDIF
        ENDIF
        ENDIF

        ;;figure out register substitution
        IFIDN <preg>,<al>
        sreg EQU <AX>
        ELSE
        IFIDN <preg>,<bl>
        sreg EQU <BX>
        ELSE
        IFIDN <preg>,<cl>
        sreg EQU <CX>
        ELSE
        IFIDN <preg>,<dl>
        sreg EQU <DX>
        ELSE
        IFIDN <preg>,<ah>
        sreg EQU <SP>
        ELSE
        IFIDN <preg>,<bh>
        sreg EQU <DI>
        ELSE
        IFIDN <preg>,<ch>
        sreg EQU <BP>

```

```

ELSE
IFIDN <preg>,<dh>
sreg EQU <SI>
ELSE
.ERR
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
;;start code generation
istart = $
        lar sreg,pmem
inext = $
isize = $-istart
iprefix = isize - dispcnt - 3
iopcode = istart + iprefix + 1
        org iopcode ;;locate for overlay opcode
        db 10H ;;umove opcode
        org inext ;;go back to next
endm

;*****
;*
;* UMOV WRITE BYTE (8 BIT) TO MEMORY FROM REGISTER, 32-BIT CS
;*
;*****
umovwr8p macro pmem,preg,padder
        local istart,inext,dispcnt,iprefix,isize,iopcode,sreg
;;figure out displacement field size
        IFIDN <padder>,<nod>
dispcnt = 0
        ELSE
        IFIDN <padder>,<d8>
dispcnt = 1
        ELSE
        IFIDN <padder>,<d16>
dispcnt = 2
        ELSE
        IFIDN <padder>,<d32>
dispcnt = 4
        ELSE
        IFIDN <padder>,<sib>
dispcnt = 1
        ELSE
        IFIDN <padder>,<sibd8>
dispcnt = 2
        ELSE
        IFIDN <padder>,<sibd32>
dispcnt = 5
        ELSE
.ERR
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF

```

```
;;figure out register substitution
```

```
IFIDN <preg>,<al>
```

```
sreg EQU <EAX>
```

```
ELSE
```

```
IFIDN <preg>,<bl>
```

```
sreg EQU <EBX>
```

```
ELSE
```

```
IFIDN <preg>,<cl>
```

```
sreg EQU <ECX>
```

```
ELSE
```

```
IFIDN <preg>,<dl>
```

```
sreg EQU <EDX>
```

```
ELSE
```

```
IFIDN <preg>,<ah>
```

```
sreg EQU <ESP>
```

```
ELSE
```

```
IFIDN <preg>,<bh>
```

```
sreg EQU <EDI>
```

```
ELSE
```

```
IFIDN <preg>,<ch>
```

```
sreg EQU <EBP>
```

```
ELSE
```

```
IFIDN <preg>,<dh>
```

```
sreg EQU <ESI>
```

```
ELSE
```

```
.ERR
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
;;start code generation
```

```
istart = $
```

```
lar sreg,pmem
```

```
inext = $
```

```
isize = $-istart
```

```
iprefix = isize - dispcnt - 3
```

```
iopcode = istart + iprefix + 1
```

```
org iopcode ;;locate for overlay opcode
```

```
db 10H ;;umove opcode
```

```
org inext ;;go back to next
```

```
endm
```



## SMM PROTECTED MODE SAMPLE

### Listing 1 Protected Mode Entry While In SMM

```

page ,132
NAME protcode
    .386P

;****extern decls ****
    EXTRN  SEG_STARTUP_REALSEG:ABS
;link time "define="
    EXTRN  _SEG_SYS_GDT_LIMIT:ABS
;GDT LIMIT
    EXTRN  _SEG_SYS_GDT_PADDR:ABS
;GDT PHYSICAL ADDRESS
    EXTRN  _SEG_SYS_IDT_LIMIT:ABS
;IDT LIMIT
    EXTRN  _SEG_SYS_IDT_PADDR:ABS
;IDT PHYSICAL ADDRESS
    EXTRN  _SEG_FLATDATA_BEGIN:PCWORD
;for flatdata selector

;***** MACROS *****
;***** MAIN PROGRAM *****

smidata segment use32 rw 'DATA' ; should be located at 60000h
;***** smi save state data
    org 0h
sscr0 dd ?
sseflg dd ?
sseip dd ?
ssedi dd ?
ssesi dd ?
ssebp dd ?
ssesp dd ?
ssebx dd ?
ssedx dd ?
ssecx dd ?
sseax dd ?

ssdr6 dd ?
ssdr7 dd ?

sstr dw ?
ssdum1 dw ?

ssldt dw ?
ssdum2 dw ?

ssgs dw ?
ssdum3 dw ?

ssfs dw ?
ssdum4 dw ?

ssds dw ?
ssdum5 dw ?

ssss dw ?
ssdum6 dw ?

sscs dw ?
ssdum7 dw ?

```

**Listing 1**
**Protected Mode Entry While In SMM (continued)**

```

sses      dw      ?
ssdum8    dw      ?

sstssa    dd      ?
sstssb    dd      ?
sstssl    dd      ?

ssidta    dd      ?
ssidtb    dd      ?
ssidtl    dd      ?

ssgdata   dd      ?
ssgdtb    dd      ?
ssgdtl    dd      ?

ssldta    dd      ?
ssldtb    dd      ?
ssldtl    dd      ?

ssgsa     dd      ?
ssgsb     dd      ?
ssgsl     dd      ?

ssfsa     dd      ?
ssfsb     dd      ?
ssfsl     dd      ?

ssdsa     dd      ?
ssdsb     dd      ?
ssdsl     dd      ?

ssssa     dd      ?
ssssb     dd      ?
ssssl     dd      ?

sscsa     dd      ?
sscsb     dd      ?
sscs1     dd      ?

ssesa     dd      ?
ssesb     dd      ?
sses1     dd      ?

          org 100h
ssptu     dd      ?
ssr22     dd      ?
ssr26     dd      ?
ssr27     dd      ?
ssr28     dd      ?
ssr29     dd      ?
ssr210    dd      ?
ssr41     dd      ?
ssr42     dd      ?
ssr43     dd      ?

          org 200h
stdata    label    byte

          org 1FFFh
lastone   db      ? ;end of 8k data segment
smidata   ends

```



**Listing 1 Protected Mode Entry While In SMM (continued)**

```

;***** stack segment *****
stack segment para STACK use32 rw 'STACK' ;locate at 62000h
stkbot db 0fffh dup (?)
stktop db ?
stack ends

;***** Protected Mode code *****
pcode segment use32 er 'CODE' ;locate at 64000h
pstart:
    assume cs:pcode,ds:smidata,ss:stack,es:nothing,
    fs:nothing,gs:nothing
    mov ebx,0F0F0F0F0h ;very large address
    mov eax,05A5A5A5AH
    ; umov es:[ebx],eax ;example write to upper memory
    db 026h,0Fh,11h,03h

;clear text screen to prove in Protected Mode
mov ebx,0B8000h ;screen address for cga style screen
mov ecx,2048 ;guess at number bytes
mov ax,0700h+' ' ;guess at char value

clrtop:
; umov es:[ebx],ax
db 066h,026h,0Fh,11h,03h
add ebx,2
loop32 clrtop

;*** Protected Mode resume
mov edi,0
mov ax,ds ;ds has smidata segment
mov es,ax
assume es:smidata
prezume db 0fh ;protect mode loadall from es:[edi]
db 07h
nop
nop
nop
nop

hlt

pcode ends

;***** startup segment *****
startup segment use16 er 'CODE'
FFFF0 equ 0ff0h ;ROM location 0fh bytes from the top
assume cs:startup ;will be set at load time

;***** CS BASED VARS *****
gdt_p label pword
dw _SEG_SYS_GDT_LIMIT
dd _SEG_SYS_GDT_PADDR
idt_p label pword
dw _SEG_SYS_IDT_LIMIT
dd _SEG_SYS_IDT_PADDR
null_p label pword
dw 0h
dd 0h

;***** ROM CODE START *****
org 100h ;bottom of 4k segment at 67000h +
100H
start:
    in al,080h ;increment port 80h
    inc al
    inc al
    out 080h,al

```

```

;***** PROTECT MODE ENTRY *****

cli
lidte cs:null_p      ;shutdown if error
lgdte cs:gdt_p       ;load gdt pointer

mov  eax,cr0          ;set protect bit
or   eax,01h
mov  cr0,eax

flush: jmp short flush ;flush prefetch queue

mov  bx,seg stack     ;setup stack
mov  ss,bx
assume ss:stack
mov  esp,offset stktop

mov  bx,seg smidata   ;setup ds
mov  ds,bx
assume ds:smidata
mov  bx,seg_SEG_FLATDATA_BEGIN ;setup es,fs,gs
mov  es,bx
mov  fs,bx
mov  gs,bx
lidte cs:idt_p        ;setup real idt
jmp  far ptr pstart   ;far intersegment jump

;***** END PROTECT MODE ENTRY *****
; REAL MODE RESUME ONLY!!!
; Setup resume from data at 60000h assumes EDI=0
mov  edi,0
mov  ax,6000h
mov  es,ax
resume db 0fh
       db 07h

nop
nop
nop
nop

hlt

;-----
;          BOOTSTRAP RESET ROUTINE
;-----
org FFFF0 ;reset location 0ffff0h remapped to (ROM size-10h)

reset:
;      jmp far ptr start
DB      0EAh ;FAR JUMP OPCODE
DW      offset start
DW      SEG_STARTUP_REALSEG
nop
nop

org FFFF0 + 000Fh
lastfill db 0AAh

startup ends          ;end of startup code

end start ;end of code, start desig is nop

```

**Listing 2 PHAR Lap Link File**

```
protcode
! object file
!-omfboot protcode.omf
!for software debugger
!-symbols
!-isymbols
!-locmap
-binary protcode.bin 060000h
!-hex protcode.hex
-386
! 386 target
-multiseg
-segsyms
-nosshide
-mapnames 20
-mapwidth 132
-build gdt,idt
-locate seg startup 067000h
-define SEG_STARTUP_REALSEG=6700h
-locate seg smidata
060000h
-locate seg stack
062000h
-locate seg pcode
064000h
-locate seg sys_gdt
063000h
-locate seg sys_idt
063100h
-segment FLATDATA limit=0FFFFFFFh rw
-locate seg FLATDATA 00000000h
```

1. The first step is to determine the type of device you are using. This is important because the pinout and voltage requirements will vary depending on the device.

2. Once you have identified the device, consult the datasheet for the specific pinout and voltage requirements. This information is typically found in the "Pin Configuration" or "Electrical Characteristics" section of the datasheet.

3. Next, determine the power requirements of the device. This includes the maximum current draw and the required voltage. Make sure that your power supply can provide the necessary power.

4. Finally, connect the device to the power supply and ground. Be careful not to connect the device to the wrong pins or to apply incorrect voltages, as this could damage the device.

5. Once the device is connected, you can begin testing it. Start by checking the output voltage and current to ensure that the device is operating correctly.

6. If you are using the device in a circuit, make sure that you have properly connected all other components and that the circuit is properly grounded.

7. Finally, once you have confirmed that the device is working correctly, you can begin using it for your intended purpose.

## **Chapter 2**

### **SCSI Products**

## CHAPTER 2

### SCSI Products

|                             |      |
|-----------------------------|------|
| Am53C94LV Data Sheet .....  | 2-3  |
| Am53CF94LV Data Sheet ..... | 2-17 |



# Am53C94LV

## Low-Voltage, High-Performance SCSI Controller

### DISTINCTIVE CHARACTERISTICS

- Functionally compatible with NCR 53C94
- Supports Low Voltage operation at 3.3 V. Conforms to JEDEC baseline specification
- AMD patented GLITCH EATER™ circuitry
- 5 MB per second SCSI transfer rate
- 20 MB per second DMA transfer rate
- 16-bit DMA Interface plus 2 bits of parity
- Flexible bus architecture, supports a three bus architecture
- Supports single ended SCSI bus
- Selection of multiplexed or non-multiplexed address and data bus
- High current drivers (48 mA) for direct connection to the single ended SCSI bus
- Supports Disconnect and Reselect commands
- Supports burst mode DMA operation with a threshold of 8
- Supports 3 byte tagged queuing as per the SCSI-2 specification
- Supports group 2 and 5 command recognition as per the SCSI-2 specification
- Advanced CMOS process for low power consumption
- Am53C94LV available in 100-pin PQFP package

### GENERAL DESCRIPTION

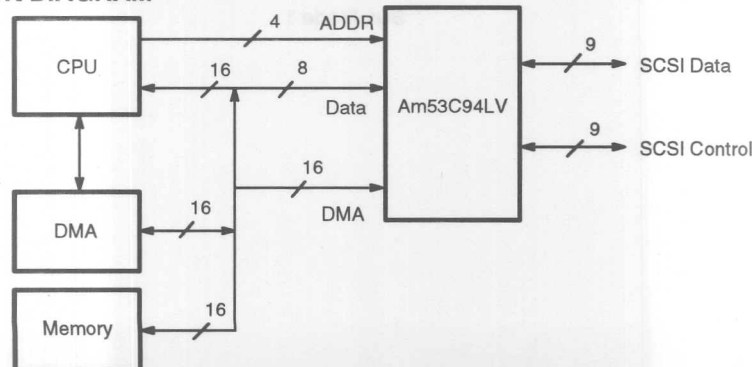
The Low-Voltage, High-Performance SCSI Controller (LVHPSC) has a flexible three bus architecture. The LVHPSC has a 16 bit DMA interface, an 8 bit host data interface and an 8 bit SCSI data interface. The LVHPSC is designed to minimize host intervention by implementing common SCSI sequences in hardware. An on-chip state machine reduces protocol overheads by performing the required sequences in response to a single command from the host. Selection, Reselection, Information Transfer and Disconnection commands are directly supported.

The 16 byte internal FIFO further assists in minimizing host involvement. The FIFO provides a temporary

storage for all command, data, status and message bytes as they are transferred between the 16 bit host data bus and the 8 bit SCSI data bus. During DMA operations the FIFO acts as a buffer to allow greater latency in the DMA channel. This permits the DMA channel to be suspended for higher priority operations such as DRAM refresh or reception of an ISDN packet.

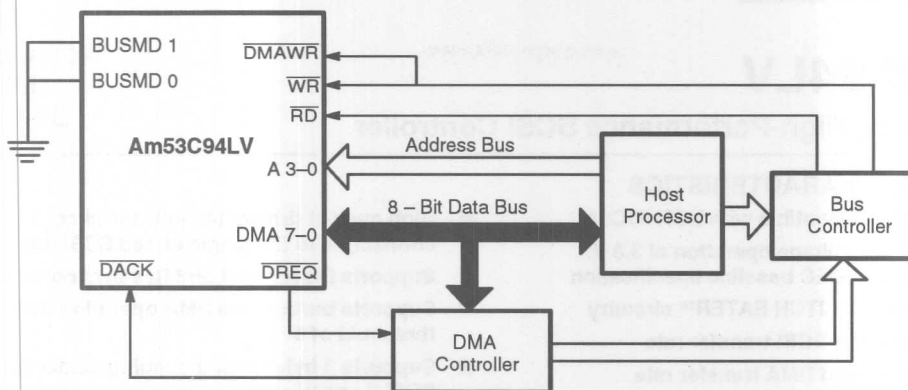
Parity on the DMA bus is optional. Parity can either be generated and checked or it can be simply passed through.

### SYSTEM BLOCK DIAGRAM



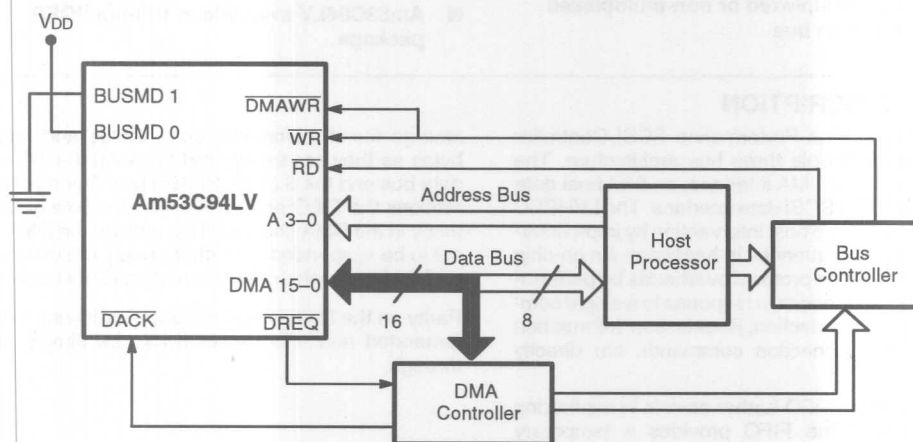
17058A-001A

SYSTEM BUS MODE DIAGRAMS



Bus Mode 0

17058A-002A

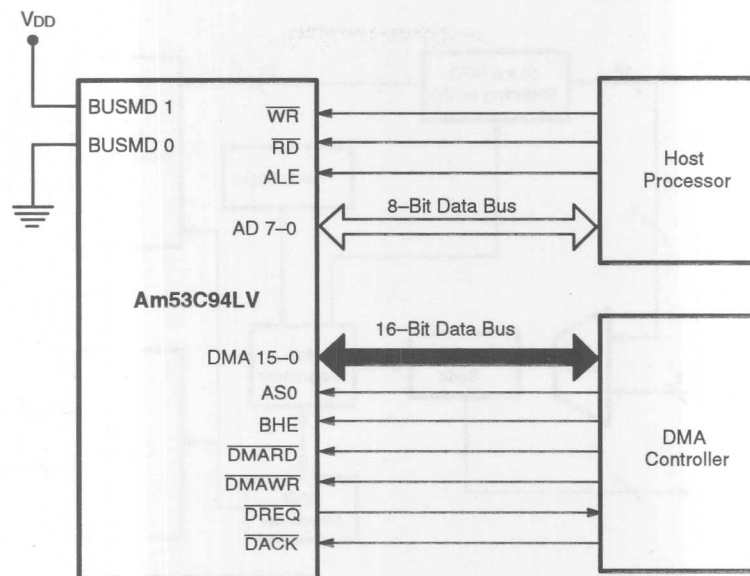


Bus Mode 1

17058A-003A

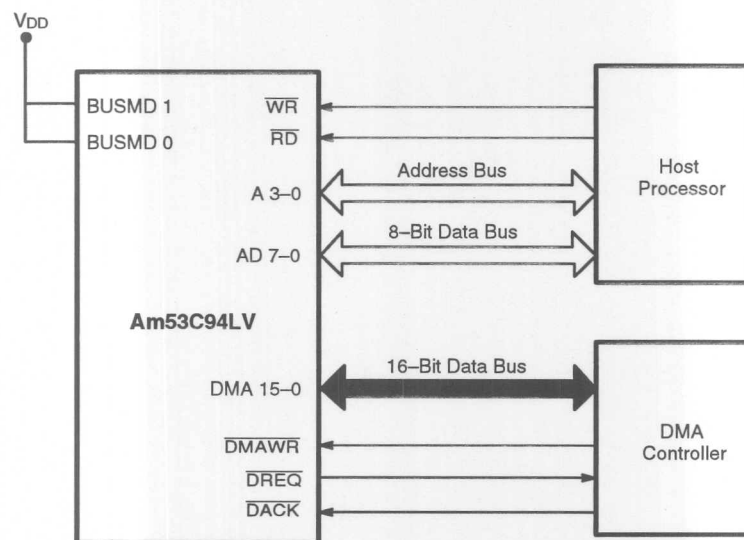


SYSTEM BUS MODE DIAGRAMS



17058A-004A

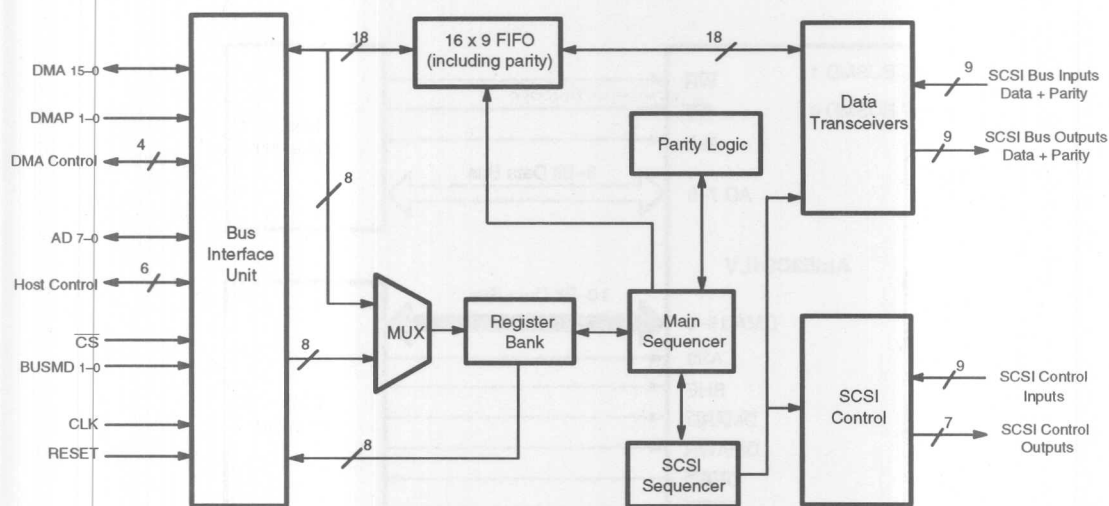
Bus Mode 2



17058A-005A

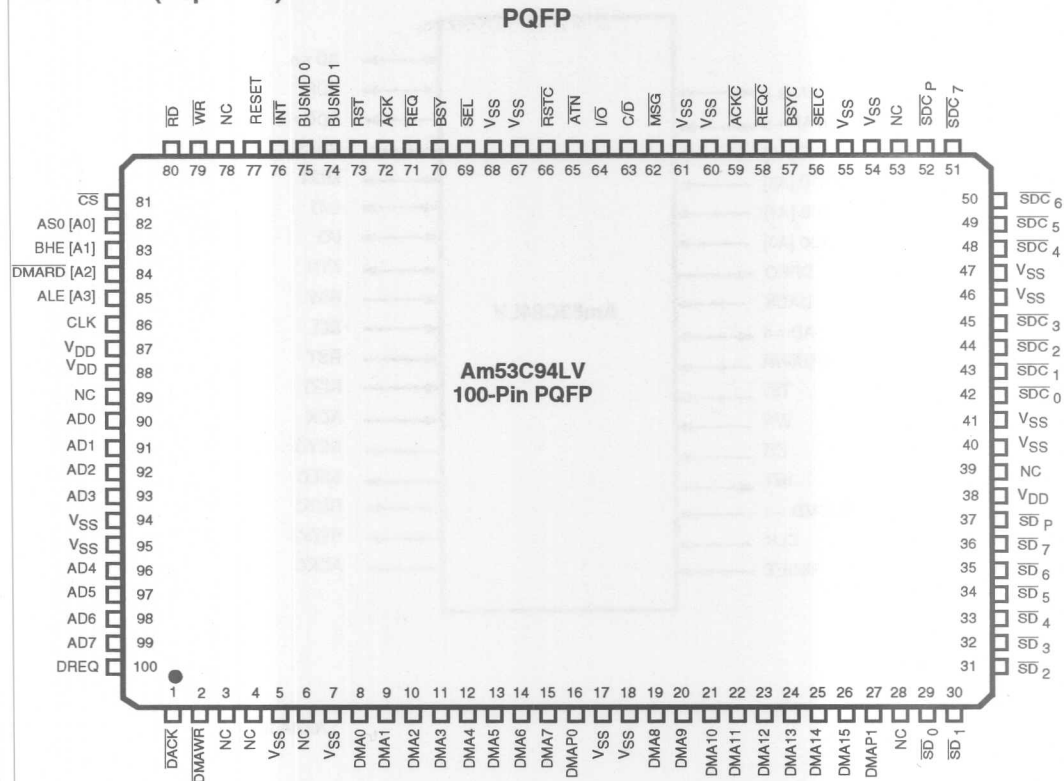
Bus Mode 3

BLOCK DIAGRAM



17058A-006A

**CONNECTION DIAGRAM**  
**Am53C94LV (Top View)**

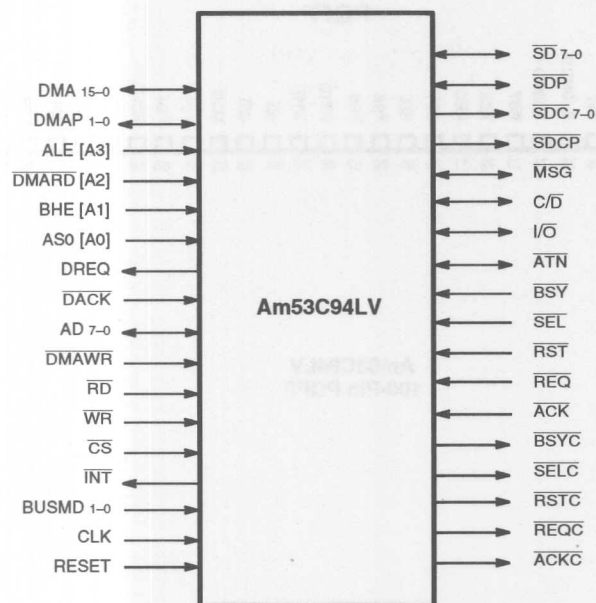


17058A-007A

**RELATED AMD PRODUCTS**

| Part Number | Description   |
|-------------|---|
| Am53C80A    | 4 MBytes/sec Asynchronous CMOS SCSI Controller                        |
| Am33C93A    | 5 MBytes/sec Async/Synchronous CMOS SCSI Controller                   |
| Am85C80     | Combination SCSI Controller (Am53C80A) and ESCC (Am85C30)             |
| Am85C30     | Enhanced Serial Communications Controller (ESCC)                      |
| Am53C94     | High-Performance CMOS SCSI Controller (Single-Ended)                  |
| Am53C96     | High-Performance CMOS SCSI Controller (Single-Ended and Differential) |
| Am53CF94    | CMOS Fast SCSI-2 Chip (Single-Ended)                                  |
| Am53CF96    | CMOS Fast SCSI-2 Controller (Single-Ended and Differential)           |
| Am53CF94LV  | Low Voltage Fast SCSI-2 Controller                                    |

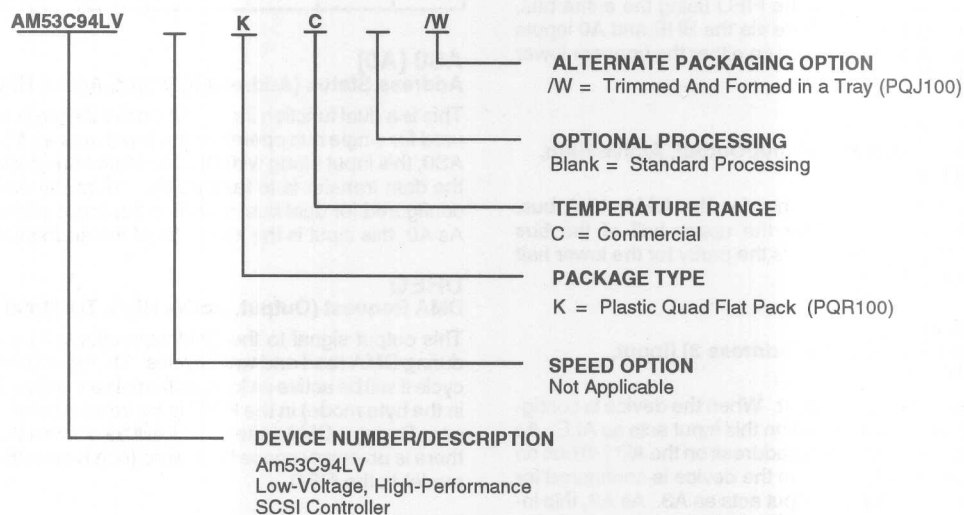
LOGIC SYMBOL



17058A-008A

**ORDERING INFORMATION****Standard Products**

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:



| Valid Combinations |          |
|--------------------|----------|
| AM53C94LV          | KC, KC/W |

**Valid Combinations**

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

## PIN DESCRIPTION

### Host Interface Signals

#### DMA 15-0

Data / DMA Bus (Input/Output, Active High, Internal Pullup)

The configuration of this bus depends on the Bus Mode 1-0 (BUSMD 1-0) inputs. When the device is configured for single bus operation, the host can access the internal register set on the lower eight lines and the DMA accesses can be made to the FIFO using the entire bus. When using the Byte Mode via the BHE and A0 inputs the data can be transferred on either the upper or lower half of the DMA 15-0 bus.

#### DMAP 1-0

Data/DMA Parity Bus (Input/Output, Active High, Internal Pullup)

These lines are odd parity for the DMA 15-0 bus. DMAP 1 is the parity for the upper half of the bus (DMA 15-8) and DMAP 0 is the parity for the lower half of the bus (DMA 7-0).

#### ALE [A3]

Address Latch Enable [Address 3] (Input, Active High)

This is a dual function input. When the device is configured for single bus operation this input acts as ALE. As ALE, this input latches the address on the AD 7-0 bus on its Low going edge. When the device is configured for dual bus operation this input acts as A3. As A3, this input is the third bit of the address bus.

#### DMARD [A2]

DMA Read [Address 2] (Input, Active Low [Active High])

This is a dual function input. When the device is configured for single bus operation this input acts as DMARD. As DMARD, this input is the read signal for the DMA 15-0 bus. When the device is configured for dual bus operation this input acts as A2. As A2, this input is the second bit of the address bus.

#### BHE [A1]

Bus High Enable [Address 1] (Input, Active High)

This is a dual function input. When the device is configured for single bus operation this input acts as BHE. As BHE, this input along with AS0 indicates on which lines the data transfer is to take place. When the device is configured for dual bus operation this input acts as A1. As A1, this input is the first bit of the address bus.

The following is the decoding for the BHE and AS0 inputs:

| BHE | AS0 | Bus Used                      |
|-----|-----|-------------------------------|
| 1   | 1   | Upper Bus – DMA 15-8, DMAP 1  |
| 1   | 0   | Full Bus – DMA 15-0, DMAP 1-0 |
| 0   | 1   | Reserved                      |
| 0   | 0   | Lower Bus – DMA 7-0, DMAP 0   |

#### AS0 [A0]

Address Status [Address 0] (Input, Active High)

This is a dual function input. When the device is configured for single bus operation this input acts as AS0. As AS0, this input along with BHE indicates on which lines the data transfer is to take place. When the device is configured for dual bus operation this input acts as A0. As A0, this input is the zeroth bit of the address bus.

#### DREQ

DMA Request (Output, Active High, Tri-State)

This output signal to the DMA controller will be active during DMA read and write cycles. During a DMA read cycle it will be active as long as there is a word (or a byte in the byte mode) in the FIFO to be transferred to memory. During a DMA write cycle it will be active as long as there is an empty space for a word (or a byte in the byte mode) in the FIFO.

#### DACK

DMA Acknowledge (Input, Active Low)

This input signal from the DMA controller will be active during DMA read and write cycles. The  $\overline{\text{DACK}}$  signal is used to access the DMA FIFO only and should never be active simultaneously with the  $\overline{\text{CS}}$  signal, which accesses the registers only.

#### AD 7-0

Host Address Data Bus (Input/Output, Active High, Internal Pullup)

This bus is used only in the dual bus mode. This bus allows the host processor to access the device's internal registers while the DMA bus is transferring data. When using the multiplexed bus mode, these lines can be used for address and data. When using a non-multiplexed bus mode these lines can be used for the data only.

#### DMAWR

DMA Write (Input, Active Low)

This signal writes the data on the DMA 15-0 bus into the internal FIFO when DACK is also active. When in the single bus mode this signal must be tied to the WR signal.

**RD****Read (Input, Active Low)**

This signal reads the internal device registers and places their contents on the data bus, when either  $\overline{CS}$  signal or  $\overline{DACK}$  signal is active.

**WR****Write (Input, Active Low)**

This signal writes the internal device registers with the value present on the data bus, when the  $\overline{CS}$  signal is also active.

**CS****Chip Select (Input, Active Low)**

This signal enables the read and write of the device registers.  $\overline{CS}$  enables access to any register (including the FIFO) while the  $\overline{DACK}$  enables access only to the FIFO.  $\overline{CS}$  and  $\overline{DACK}$  should never be active simultaneously in the single bus mode, they may however be active simultaneously in the dual bus mode provided the  $\overline{CS}$  signal is not enabling access to the FIFO.

**INT****Interrupt (Output, Active Low, Open Drain)**

This signal is a non-maskable interrupt flag to the host processor. This signal is latched on the output on the high going edge of the clock. This flag may be cleared by reading the Interrupt Status Register (ISTAT) or by performing a device reset (hard or soft). This flag is not cleared by a SCSI reset.

**BUSMD 1-0****Bus Mode (Input, Active High)**

These inputs configure the device for single bus or dual bus operation and the DMA width.

| BUSMD1 | BUSMD0 | Bus Configuration  |
|--------|--------|--|
| 1      | 1      | Two buses: 8-bit Host Bus & 16-bit DMA Bus<br>Register Address on A 3-0 & Data on AD Bus   |
| 1      | 0      | Two buses: Multiplexed & byte control<br>Register Address on AD 3-0 & Data on AD Bus       |
| 0      | 1      | Single bus: 8-bit Host Bus & 16-bit DMA Bus<br>Register Address on A 3-0 & Data on DMA Bus |
| 0      | 0      | Single bus: 8-bit Host Bus & 8-bit DMA Bus<br>Register Address on A 3-0 & Data on DMA Bus  |

**CLK****Clock (Input)**

Clock input used to generate all the internal device timings. The maximum frequency of this input is 25 MHz. A minimum of 10MHz is required to maintain the SCSI bus timings.

**RESET****Reset (Input, Active High)**

This input when active resets the device. The RESET input must be active for at least two CLK periods after the voltage on the power inputs has reached  $V_{cc}$  minimum.

**SCSI Interface Signals****SD 7-0****SCSI Data (Input, Active Low, Schmitt Trigger)**

These are SCSI data input pins.

**SDP****SCSI Data Parity (Input, Active Low, Schmitt Trigger)**

This is the SCSI data parity input pin.

**SDC 7-0****SCSI Data Control (Output, Active Low, Open Drain)**

These are SCSI data output pins.

**SDCP****SCSI Data Control Parity (Output, Active Low, Open Drain)**

This is the SCSI data parity output pin.

**MSG****Message (Input/Output, Active Low, Schmitt Trigger)**

This is a bidirectional signal with a 48 mA output driver. It is an output in the target mode and a Schmitt trigger input in the initiator mode.

**C/D****Command/Data (Input/Output, Schmitt Trigger)**

This is a bidirectional signal with a 48 mA output driver. It is an output in the target mode and a Schmitt trigger input in the initiator mode.

**I/O****Input/Output (Input/Output, Schmitt Trigger)**

This is a bidirectional signal with a 48 mA output driver. It is an output in the target mode and a Schmitt trigger input in the initiator mode.

### ATN

**Attention (Input/Output, Active Low, Schmitt Trigger)**

This signal is a 48 mA output in the initiator mode and a Schmitt trigger input in the target mode. This signal will be asserted when the device detects a parity error or it can be asserted via certain commands. In the target mode this pin is an input.

### BSY

**Busy (Input, Active Low, Schmitt Trigger)**

This is a SCSI input signal with a Schmitt trigger.

### SEL

**Select (Input, Active Low, Schmitt Trigger)**

This is a SCSI input signal with a Schmitt trigger.

### RST

**Reset (Input, Active Low, Schmitt Trigger)**

This is a SCSI input signal with a Schmitt trigger.

### REQ

**Request (Input, Active Low, Schmitt Trigger)**

This is a SCSI input signal with a Schmitt trigger.

### ACK

**Acknowledge (Input, Active Low, Schmitt Trigger).** This is a SCSI input signal with a Schmitt trigger.

### BSYC

**Busy Control (Output, Active Low, Open Drain)**

This is a SCSI output with 48 mA drive. This pin is the BSY output for the SCSI bus.

### SEL<sub>C</sub>

**Select Control (Output, Active Low, Open Drain)**

This is a SCSI output with 48 mA drive. This pin is the SEL output for the SCSI bus.

### RST<sub>C</sub>

**Reset Control (Output, Active Low, Open Drain)**

This is a SCSI output with 48 mA drive. The Reset SCSI command will cause the device to drive RST<sub>C</sub> active for 25–40 microseconds, which will depend on the CLK frequency and the conversion factor. This pin is the RST output for the SCSI bus.

### REQ<sub>C</sub>

**Request Control (Output, Active Low, Open Drain)**

This is a SCSI output with 48 mA drive. This signal is asserted only in the target mode.

### ACK<sub>C</sub>

**Acknowledge Control (Output, Active Low, Open Drain)**

This is a SCSI output with 48 mA drive. This signal is asserted only in the initiator mode.



## FUNCTIONAL DESCRIPTION

## Register Map

| Address (Hex.) | Operation  | Register                             |
|----------------|------------|--------------------------------------|
| 00             | Read       | Current Transfer Count Register LSB  |
| 00             | Write      | Start Transfer Count Register LSB    |
| 01             | Read       | Current Transfer Count Register MSB  |
| 01             | Write      | Start Transfer Count Register MSB    |
| 02             | Read/Write | FIFO Register                        |
| 03             | Read/Write | Command Register                     |
| 04             | Read       | Status Register                      |
| 04             | Write      | SCSI Destination ID Register         |
| 05             | Read       | Interrupt Status Register            |
| 05             | Write      | SCSI Timeout Register                |
| 06             | Read       | Internal State Register              |
| 06             | Write      | Synchronous Transfer Period Register |
| 07             | Read       | Current FIFO Internal State Register |
| 07             | Write      | Synchronous Offset Register          |
| 08             | Read/Write | Control Register 1                   |
| 09             | Write      | Clock Factor Register                |
| 0A             | Write      | Forced Test Mode Register            |
| 0B             | Read/Write | Control Register 2                   |
| 0C             | Read/Write | Control Register 3 Rev. ID Register  |
| 0F             | Write      | Data Alignment Register              |

Not all registers in this device are both readable and writable. Some read only registers share the same address with write only registers. The registers can be accessed by asserting the  $\overline{CS}$  signal and then asserting either  $\overline{RD}$  or  $\overline{WR}$  signal depending on the operation to be performed. Only the FIFO Register can be accessed by asserting either  $\overline{CS}$  or  $\overline{DACK}$  in conjunction with  $\overline{RD}$  and  $\overline{WR}$  signals or  $\overline{DMARD}$  and  $\overline{DMAWR}$  signals. The register address inputs are ignored when  $\overline{DACK}$  is used but must be valid when  $\overline{CS}$  is used.

## COMMANDS

The device commands can be broadly divided into two categories, DMA commands and non-DMA commands. DMA commands are those which cause data movement between the host memory and the SCSI bus while non-DMA commands are those that cause data movement between the device FIFO and the SCSI bus. The MSB of the command byte differentiates DMA commands from non-DMA commands.

### Summary of Commands

| Command                          | Command Code (Hex.) |          |
|----------------------------------|---------------------|----------|
|                                  | Non-DMA Mode        | DMA Mode |
| <b>Initiator Commands</b>        |                     |          |
| Information Transfer             | 10                  | 90       |
| Initiator Command Complete Steps | 11                  | 91       |
| Message Accepted                 | 12                  | -        |
| Transfer Pad Bytes               | 18                  | 98       |
| Set $\overline{\text{ATN}}$      | 1A                  | -        |
| Reset $\overline{\text{ATN}}$    | 1B                  | -        |
| <b>Target Commands</b>           |                     |          |
| Send Message                     | 20                  | A0       |
| Send Status                      | 21                  | A1       |
| Send Data                        | 22                  | A2       |
| Disconnect Steps                 | 23                  | A3       |
| Terminate Steps                  | 24                  | A4       |
| Target Command Complete Steps    | 25                  | A5       |
| Disconnect                       | 27                  | A7       |
| Receive Message                  | 28                  | A8       |
| Receive Command Steps            | 29                  | A9       |
| Receive Data                     | 2A                  | AA       |
| Receive Command Steps            | 2B                  | AB       |
| Target Abort DMA                 | 04                  | 84       |

| Command  | Command Code (Hex.) |          |
|--|---------------------|----------|
|  | Non-DMA Mode        | DMA Mode |
| <b>Idle State Commands</b>                         |                     |          |
| Reselect Steps                                     | 40                  | C0       |
| Select without $\overline{\text{ATN}}$ Steps       | 41                  | C1       |
| Select with $\overline{\text{ATN}}$ Steps          | 42                  | C2       |
| Select with $\overline{\text{ATN}}$ and Stop Steps | 43                  | C3       |
| Enable Selection/Reselection                       | 44                  | C4       |
| Disable Selection/Reselection                      |                     | 45       |
| Select with $\overline{\text{ATN3}}$               | 46                  | C6       |
| <b>General Commands</b>                            |                     |          |
| No Operation                                       | 00                  | 80       |
| Clear FIFO   | 01                  | 81       |
| Reset Device                                       | 02                  | 82       |
| Reset SCSI bus                                     | 03                  | 83       |

**ABSOLUTE MAXIMUM RATINGS**

Storage Temperature . . . . . -55 to +125°C  
 Ambient Operating Temperature . . . . . 0 to +70°C  
 Maximum  $V_{CC}$  . . . . . -0.5 to +7.0 V  
 DC Voltage Applied to Any Pin . . -0.5 to ( $V_{DD} + 0.3$ ) V  
 Input Static Discharge Protection . . 3000 V pin to pin  
 (Human body model: 100 pF at 1.5 K $\Omega$ )

*Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.*

**OPERATING RANGES****Commercial (C) Devices**

Ambient Temperature ( $T_A$ ) . . . . . 0 to +70°C  
 Supply Voltage ( $V_{DD}$ ) . . . . . 3.3 V  $\pm$ 10%

*Operating ranges define those limits between which the functionality of the device is guaranteed.*

**DC OPERATING CHARACTERISTICS**  $V_{DD}$ =2.9 V to 3.7 V;  $T_{CASE}$  = 0° to +100°

| Parameter Symbol          | Parameter Description   | Pin Names  | Test Condition                          | Min.     | Max.           | Unit          |
|---------------------------|-------------------------|--|---|----------|----------------|---------------|
| $I_{CCS}$                 | Static Supply Current   |  |   |          | 4.0            | mA            |
| $I_{CCD}$                 | Dynamic Supply Current  |  |   |          | 50             | mA            |
| $I_{LU}$                  | Latch Up Current        |  |   | -100     | +100           | mA            |
| <b>SCSI Pins</b>          |                         |  |   |          |                |               |
| $V_{IH}$                  | Input High Voltage      | All SCSI Inputs  |   | 2.0      | $V_{DD} + 0.3$ | V             |
| $V_{IL}$                  | Input Low Voltage       | All SCSI Inputs  |   | -0.3     | 0.8            | V             |
| $V_{IHST}$                | Input Hysteresis        | All SCSI Inputs  | $2.9\text{ V} < V_{DD} < 3.7\text{ V}$  | 200      |                | mV            |
| $V_{OH}$                  | Output High Voltage     | SD 7-0, SDP  |   |          |                |               |
| $V_{SOL1}$                | SCSI Output Low Voltage | SD 7-0, SDP  | $I_{OL} = 4\text{ mA}$                  | $V_{SS}$ | 0.4            | V             |
| $V_{SOL2}$                | SCSI Output Low Voltage | SDC 7-0, SDCP, MSG, C/D, I/O, ATN, RSTC, SELC, BSYC, ACKC and REQC | $I_{OL} = 48\text{ mA}$                 | $V_{SS}$ | 0.5            | V             |
| $I_{LI}$                  | Input Leakage           |  | $0.0\text{ V} \leq V_{IN} \leq V_{DD}$  | -10      | +10            | $\mu\text{A}$ |
| $I_{LO}$                  | Output Leakage          |  | $0.1\text{ V} \leq V_{OUT} \leq V_{DD}$ | -10      | +10            | $\mu\text{A}$ |
| <b>Bidirectional Pins</b> |                         |  |   |          |                |               |
| $V_{IH}$                  | Input High Voltage      |  |   | 2.0      | $V_{DD} + 0.3$ | V             |
| $V_{IL}$                  | Input Low Voltage       |  |   | -0.3     | 0.8            | V             |
| $V_{OH}$                  | Output High Voltage     | DMA 15-0, DMAP 1-0 and AD 7-0                                      |   |          |                |               |
| $V_{OL}$                  | Output Low Voltage      | DMA 15-0, DMAP 1-0 and AD 7-0                                      | $I_{OL} = 0.5\text{ mA}$                | $V_{SS}$ | 0.2            | V             |
|                           |                         |  | $I_{OL} = 2\text{ mA}$                  |          | 0.45           | V             |
| $I_{LI}$                  | Input Leakage           | DMA 15-0, DMAP 1-0 and AD 7-0                                      | $0.0\text{ V} \leq V_{IN} \leq V_{DD}$  |          | $\pm 15$       | $\mu\text{A}$ |
| $I_{LO}$                  | Output Leakage          | DMA 15-0, DMAP 1-0 and AD 7-0                                      | $0.1\text{ V} \leq V_{OUT} \leq V_{DD}$ |          | $\pm 15$       | $\mu\text{A}$ |
| <b>Output Pins</b>        |                         |  |   |          |                |               |
| $V_{OH}$                  | Output High Voltage     | DRQ and INT  |   |          |                |               |
| $V_{OL}$                  | Output Low Voltage      | DRQ and INT  | $I_{OL} = 0.5\text{ mA}$                |          | 0.2            | V             |
|                           |                         |  | $I_{OL} = 2.5\text{ mA}$                |          | 0.45           | V             |



## Am53C94LV Data Sheet



# Am53CF94LV

## Low-Voltage Fast SCSI-2 Controller

### DISTINCTIVE CHARACTERISTICS

- Functionally compatible with Emulex FAS216 and NCR 53CF94
- Supports Low-Voltage operation at 3.3 V. Conforms to JEDEC baseline specifications
- AMD patented GLITCH EATER™ circuitry
- 10 MB per second SCSI transfer rate
- 20 MB per second DMA transfer rate
- 16-bit DMA Interface plus 2 bits of parity
- Flexible bus architecture, supports a three bus architecture
- Supports single ended SCSI bus
- Selection of multiplexed or non-multiplexed address and data bus
- High current drivers (48 mA) for direct connection to the single ended SCSI bus
- Supports Disconnect and Reselect commands
- Supports burst mode DMA operation with a threshold of 8
- Supports 3 byte tagged queuing as per the SCSI-2 specification
- Supports group 2 and 5 command recognition as per the SCSI-2 specification
- Advanced CMOS process for low power consumption
- Am53CF94LV available in 100-pin PQFP package

### GENERAL DESCRIPTION

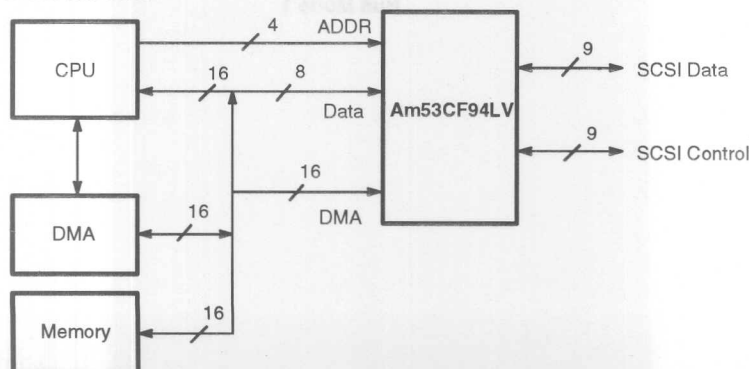
The Low Voltage Fast SCSI-2 Controller (LVFSC) has a flexible three bus architecture. The LVFSC has a 16 bit DMA interface, an 8 bit host data interface and an 8 bit SCSI data interface. The LVFSC is designed to minimize host intervention by implementing common SCSI sequences in hardware. An on-chip state machine reduces protocol overheads by performing the required sequences in response to a single command from the host. Selection, Reselection, Information Transfer and Disconnection commands are directly supported.

The 16 byte internal FIFO further assists in minimizing host involvement. The FIFO provides a temporary

storage for all command, data, status and message bytes as they are transferred between the 16 bit host data bus and the 8 bit SCSI data bus. During DMA operations the FIFO acts as a buffer to allow greater latency in the DMA channel. This permits the DMA channel to be suspended for higher priority operations such as DRAM refresh or reception of an ISDN packet.

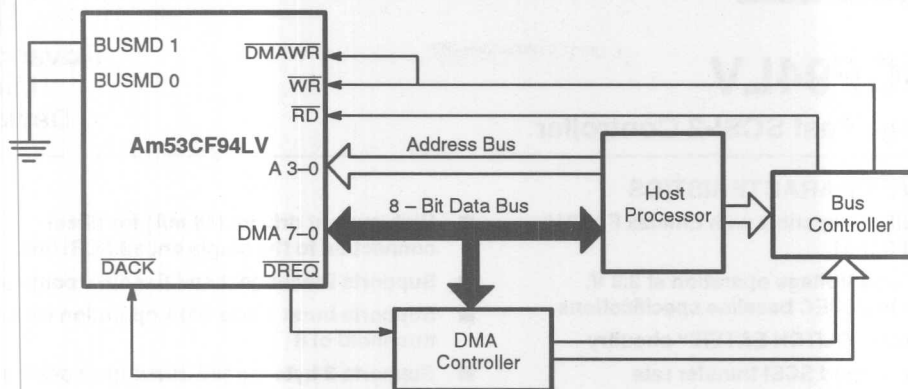
Parity on the DMA bus is optional. Parity can either be generated and checked or it can be simply passed through.

### SYSTEM BLOCK DIAGRAM



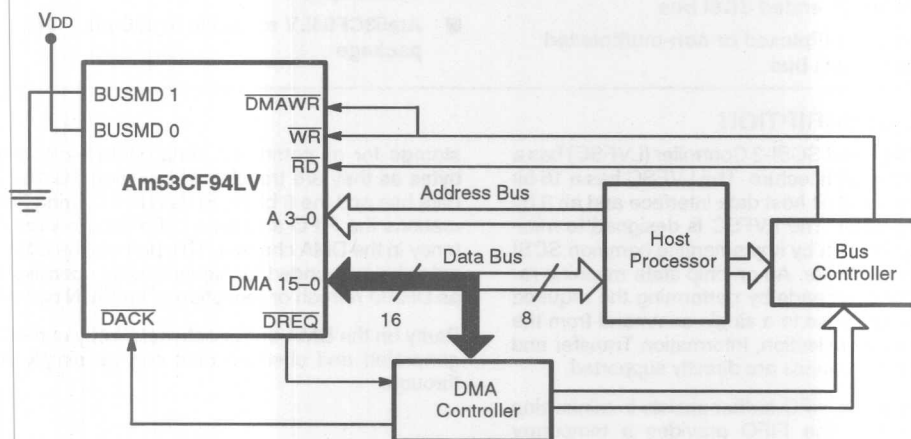
17059A-001A

SYSTEM BUS MODE DIAGRAMS



Bus Mode 0

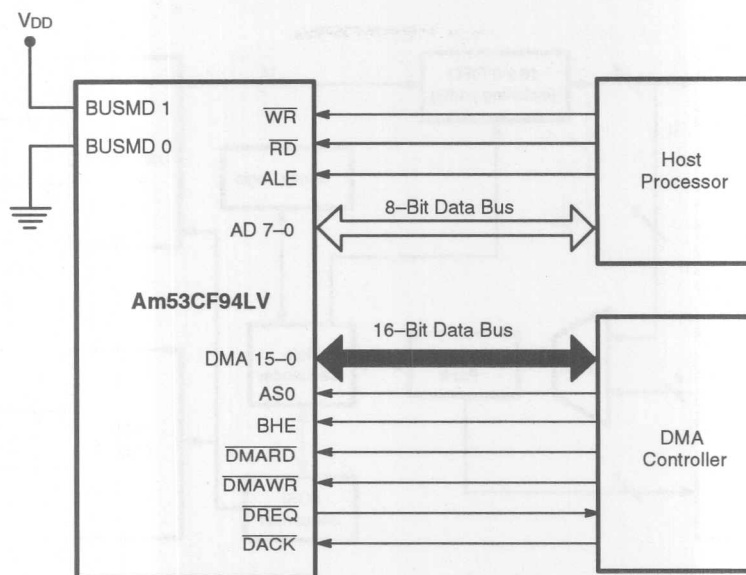
17059A-002A



Bus Mode 1

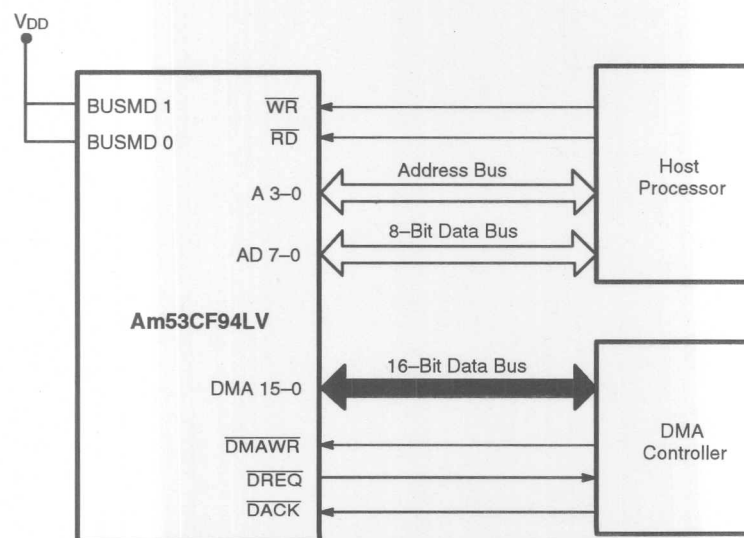
17059A-003A

SYSTEM BUS MODE DIAGRAMS



17059A-004A

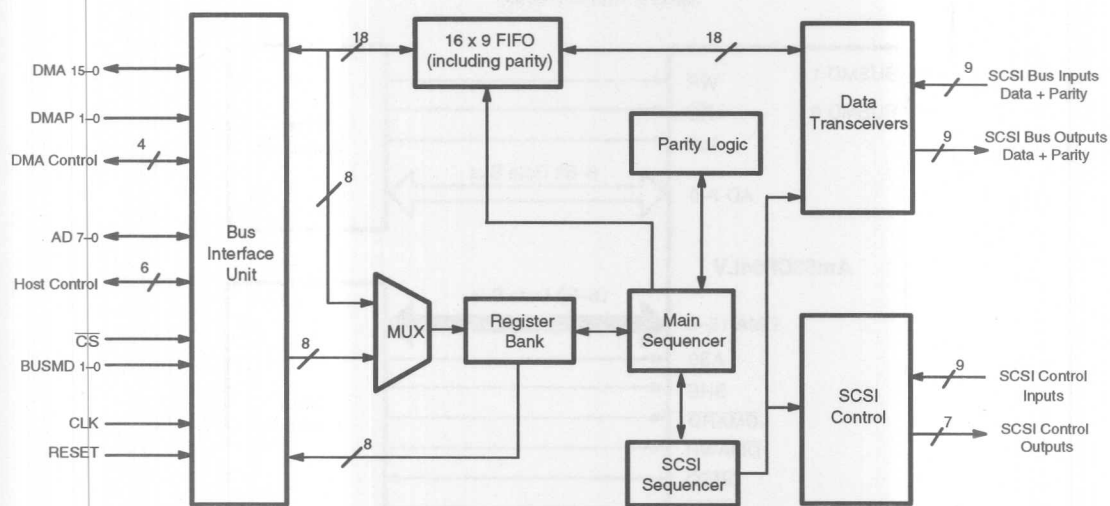
Bus Mode 2



17059A-005A

Bus Mode 3

BLOCK DIAGRAM

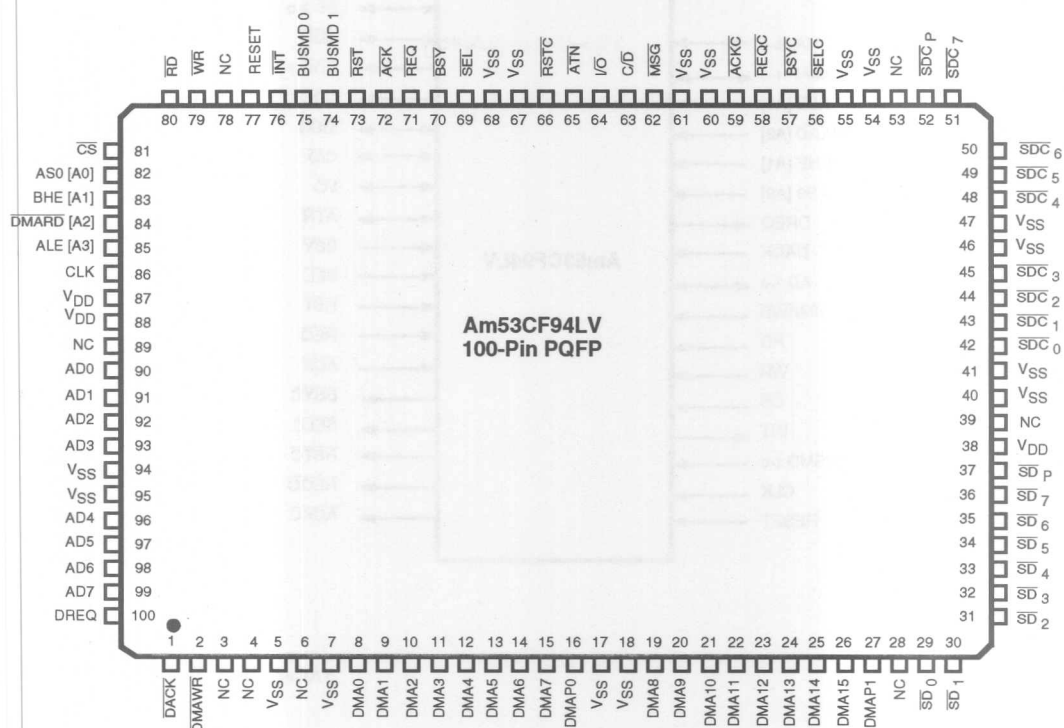




## CONNECTION DIAGRAM

## Am53CF94LV (Top View)

PQFP

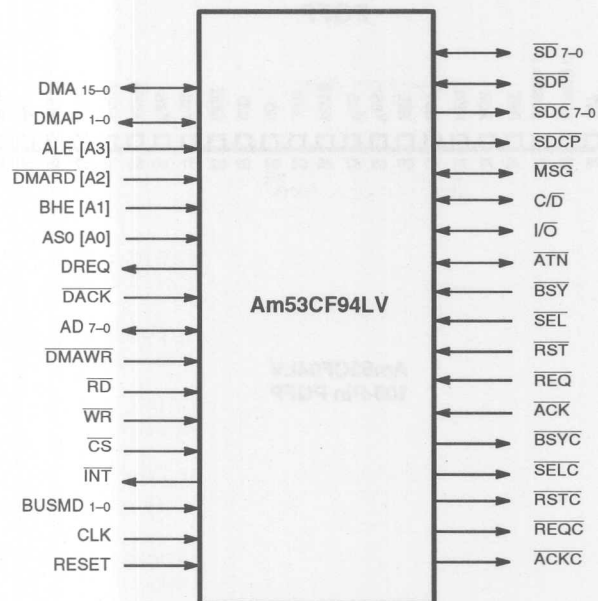


17059A-007A

## RELATED AMD PRODUCTS

| Part Number | Description   |
|-------------|---|
| Am53C80A    | 4 MBytes/sec Asynchronous CMOS SCSI Controller                        |
| Am33C93A    | 5 MBytes/sec Async/Synchronous CMOS SCSI Controller                   |
| Am85C80     | Combination SCSI Controller (Am53C80A) and ESCC (Am85C30)             |
| Am85C30     | Enhanced Serial Communications Controller (ESCC)                      |
| Am53C94     | High-Performance CMOS SCSI Controller (Single-Ended)                  |
| Am53C96     | High-Performance CMOS SCSI Controller (Single-Ended and Differential) |
| Am53C94LV   | Low-Voltage High-Performance SCSI Controller                          |
| Am53CF94    | CMOS Fast SCSI-2 Controller (Single-Ended)                            |
| Am53CF96    | CMOS Fast SCSI-2 Controller (Single-Ended and Differential)           |

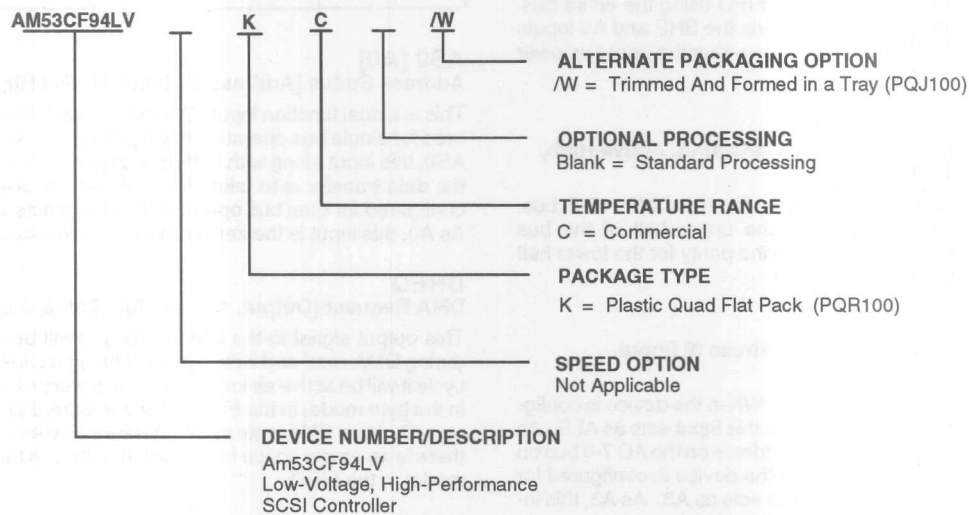
LOGIC SYMBOL



17059A-008A

**ORDERING INFORMATION****Standard Products**

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:



| Valid Combinations |          |
|--------------------|----------|
| AM53CF94LV         | KC, KC/W |

**Valid Combinations**

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

## **DMA 15-0**

### **Data/DMA Bus (Input/Output, Active High, Internal Pullup)**

The configuration of this bus depends on the Bus Mode 1-0 (BUSMD 1-0) inputs. When the device is configured for single bus operation, the host can access the internal register set on the lower eight lines and the DMA accesses can be made to the FIFO using the entire bus. When using the Byte Mode via the BHE and A0 inputs the data can be transferred on either the upper or lower half of the DMA 15-0 bus.

## **DMAP 1-0**

### **Data/DMA Parity Bus (Input/Output, Active High, Internal Pullup)**

These lines are odd parity for the DMA 15-0 bus. DMAP 1 is the parity for the upper half of the bus (DMA 15-8) and DMAP 0 is the parity for the lower half of the bus (DMA 7-0).

## **ALE [A3]**

### **Address Latch Enable [Address 3] (Input, Active High)**

This is a dual function input. When the device is configured for single bus operation this input acts as ALE. As ALE, this input latches the address on the AD 7-0 bus on its Low going edge. When the device is configured for dual bus operation this input acts as A3. As A3, this input is the third bit of the address bus.

## **DMARD [A2]**

### **DMA Read [Address 2] (Input, Active Low [Active High])**

This is a dual function input. When the device is configured for single bus operation this input acts as DMARD. As DMARD, this input is the read signal for the DMA 15-0 bus. When the device is configured for dual bus operation this input acts as A2. As A2, this input is the second bit of the address bus.

## **BHE [A1]**

### **Bus High Enable [Address 1] (Input, Active High)**

This is a dual function input. When the device is configured for single bus operation this input acts as BHE. As BHE, this input along with AS0 indicates on which lines the data transfer is to take place. When the device is configured for dual bus operation this input acts as A1. As A1, this input is the first bit of the address bus.

inputs:

| BHE | AS0 | Bus Used                      |
|-----|-----|-------------------------------|
| 1   | 1   | Upper Bus – DMA 15-8, DMAP 1  |
| 1   | 0   | Full Bus – DMA 15-0, DMAP 1-0 |
| 0   | 1   | Reserved                      |
| 0   | 0   | Lower Bus – DMA 7-0, DMAP 0   |

## **AS0 [A0]**

### **Address Status [Address 0] (Input, Active High)**

This is a dual function input. When the device is configured for single bus operation this input acts as AS0. As AS0, this input along with BHE indicates on which lines the data transfer is to take place. When the device is configured for dual bus operation this input acts as A0. As A0, this input is the zeroth bit of the address bus.

## **DREQ**

### **DMA Request (Output, Active High, Three-State)**

This output signal to the DMA controller will be active during DMA read and write cycles. During a DMA read cycle it will be active as long as there is a word (or a byte in the byte mode) in the FIFO to be transferred to memory. During a DMA write cycle it will be active as long as there is an empty space for a word (or a byte in the byte mode) in the FIFO.

## **DACK**

### **DMA Acknowledge (Input, Active Low)**

This input signal from the DMA controller will be active during DMA read and write cycles. The DACK signal is used to access the DMA FIFO only and should never be active simultaneously with the CS signal, which accesses the registers only.

## **AD 7-0**

### **Host Address Data Bus (Input/Output, Active High, Internal Pullup)**

This bus is used only in the dual bus mode. This bus allows the host processor to access the device's internal registers while the DMA bus is transferring data. When using the multiplexed bus mode, these lines can be used for address and data. When using a non-multiplexed bus mode these lines can be used for the data only.

## **DMAWR**

### **DMA Write (Input, Active Low)**

This signal writes the data on the DMA 15-0 bus into the internal FIFO when DACK is also active. When in the single bus mode this signal must be tied to the WR signal.

**RD****Read (Input, Active Low)**

This signal reads the internal device registers and places their contents on the data bus, when either  $\overline{CS}$  signal or  $\overline{DACK}$  signal is active.

**WR****Write (Input, Active Low)**

This signal writes the internal device registers with the value present on the data bus, when the  $\overline{CS}$  signal is also active.

 **$\overline{CS}$** **Chip Select (Input, Active Low)**

This signal enables the read and write of the device registers.  $\overline{CS}$  enables access to any register (including the FIFO) while the  $\overline{DACK}$  enables access only to the FIFO.  $\overline{CS}$  and  $\overline{DACK}$  should never be active simultaneously in the single bus mode, they may however be active simultaneously in the dual bus mode provided the  $\overline{CS}$  signal is not enabling access to the FIFO.

**INT****Interrupt (Output, Active Low, Open Drain)**

This signal is a non-maskable interrupt flag to the host processor. This signal is latched on the output on the high going edge of the clock. This flag may be cleared by reading the Interrupt Status Register (ISTAT) or by performing a device reset (hard or soft). This flag is not cleared by a SCSI reset.

**BUSMD 1-0****Bus Mode (Input, Active High)**

These inputs configure the device for single bus or dual bus operation and the DMA width.

| BUSMD1 | BUSMD0 | Bus Configuration  |
|--------|--------|--|
| 1      | 1      | Two buses: 8-bit Host Bus & 16-bit DMA Bus<br>Register Address on A 3-0 & Data on AD Bus   |
| 1      | 0      | Two buses: Multiplexed & byte control<br>Register Address on AD 3-0 & Data on AD Bus       |
| 0      | 1      | Single bus: 8-bit Host Bus & 16-bit DMA Bus<br>Register Address on A 3-0 & Data on DMA Bus |
| 0      | 0      | Single bus: 8-bit Host Bus & 8-bit DMA Bus<br>Register Address on A 3-0 & Data on DMA Bus  |

**CLK****Clock (Input)**

Clock input used to generate all the internal device timings. The maximum frequency of this input is 25 MHz. A minimum of 10MHz is required to maintain the SCSI bus timings.

**RESET****Reset (Input, Active High)**

This input when active resets the device. The RESET input must be active for at least two CLK periods after the voltage on the power inputs has reached  $V_{cc}$  minimum.

**SCSI Interface Signals** **$\overline{SD} 7-0$** **SCSI Data (Input, Active Low, Schmitt Trigger)**

These are SCSI data input pins.

 **$\overline{SDP}$** **SCSI Data Parity (Input, Active Low, Schmitt Trigger)**

This is the SCSI data parity input pin.

 **$\overline{SDC} 7-0$** **SCSI Data Control (Output, Active Low, Open Drain)**

These are SCSI data output pins.

 **$\overline{SDCP}$** **SCSI Data Control Parity (Output, Active Low, Open Drain)**

This is the SCSI data parity output pin.

**MSG****Message (Input/Output, Active Low, Schmitt Trigger)**

This is a bidirectional signal with a 48 mA output driver. It is an output in the target mode and a Schmitt trigger input in the initiator mode.

 **$C/\overline{D}$** **Command/Data (Input/Output, Schmitt Trigger)**

This is a bidirectional signal with a 48 mA output driver. It is an output in the target mode and a Schmitt trigger input in the initiator mode.

 **$I/\overline{O}$** **Input/Output (Input/Output, Schmitt Trigger)**

This is a bidirectional signal with a 48 mA output driver. It is an output in the target mode and a Schmitt trigger input in the initiator mode.

## ATN

**Attention (Input/Output, Active Low, Schmitt Trigger)**

This signal is a 48 mA output in the initiator mode and a Schmitt trigger input in the target mode. This signal will be asserted when the device detects a parity error or it can be asserted via certain commands. In the target mode this pin is an input.

## BSY

**Busy (Input, Active Low, Schmitt Trigger)**

This is a SCSI input signal with a Schmitt trigger.

## SEL

**Select (Input, Active Low, Schmitt Trigger)**

This is a SCSI input signal with a Schmitt trigger.

## RST

**Reset (Input, Active Low, Schmitt Trigger)**

This is a SCSI input signal with a Schmitt trigger.

## REQ

**Request (Input, Active Low, Schmitt Trigger)**

This is a SCSI input signal with a Schmitt trigger.

## ACK

**Acknowledge (Input, Active Low, Schmitt Trigger).** This is a SCSI input signal with a Schmitt trigger.

## BSYC

**Busy Control (Output, Active Low, Open Drain)**

This is a SCSI output with 48 mA drive. This pin is the BSY output for the SCSI bus.

## SELC

**Select Control (Output, Active Low, Open Drain)**

This is a SCSI output with 48 mA drive. When the device is configured in the Single-Ended SCSI Mode (DFMODE inactive), this pin is the SEL output for the SCSI bus.

## RSTC

**Reset Control (Output, Active Low, Open Drain)**

This is a SCSI output with 48 mA drive. The Reset SCSI command will cause the device to drive RSTC active for 25–40 microseconds, which will depend on the CLK frequency and the conversion factor. This pin is the RST output for the SCSI bus.

## REQC

**Request Control (Output, Active Low, Open Drain)**

This is a SCSI output with 48 mA drive. This signal is asserted only in the target mode.

## ACKC

**Acknowledge Control (Output, Active Low, Open Drain)**

This is a SCSI output with 48 mA drive. This signal is asserted only in the initiator mode.

## FUNCTIONAL DESCRIPTION

## Register Map

| Address (Hex.) | Operation  | Register                             |
|----------------|------------|--------------------------------------|
| 00             | Read       | Current Transfer Count Register LSB  |
| 00             | Write      | Start Transfer Count Register LSB    |
| 01             | Read       | Current Transfer Count Register MSB  |
| 01             | Write      | Start Transfer Count Register MSB    |
| 02             | Read/Write | FIFO Register                        |
| 03             | Read/Write | Command Register                     |
| 04             | Read       | Status Register                      |
| 04             | Write      | SCSI Destination ID Register         |
| 05             | Read       | Interrupt Status Register            |
| 05             | Write      | SCSI Timeout Register                |
| 06             | Read       | Internal State Register              |
| 06             | Write      | Synchronous Transfer Period Register |
| 07             | Read       | Current FIFO Internal State Register |
| 07             | Write      | Synchronous Offset Register          |
| 08             | Read/Write | Control Register 1                   |
| 09             | Write      | Clock Factor Register                |
| 0A             | Write      | Forced Test Mode Register            |
| 0B             | Read/Write | Control Register 2                   |
| 0C             | Read/Write | Control Register 3 Rev. ID Register  |
| 0F             | Write      | Data Alignment Register              |

Not all registers in this device are both readable and writable. Some read only registers share the same address with write only registers. The registers can be accessed by asserting the  $\overline{CS}$  signal and then asserting either  $\overline{RD}$  or  $\overline{WR}$  signal depending on the operation to be performed. Only the FIFO Register can be accessed by asserting either  $\overline{CS}$  or  $\overline{DACK}$  in conjunction with  $\overline{RD}$  and  $\overline{WR}$  signals or  $\overline{DMARD}$  and  $\overline{DMAWR}$  signals. The register address inputs are ignored when  $\overline{DACK}$  is used but must be valid when  $\overline{CS}$  is used.



## COMMANDS

The device commands can be broadly divided into two categories, DMA commands and non-DMA commands. DMA commands are those which cause data movement between the host memory and the SCSI bus while non-DMA commands are those that cause data movement between the device FIFO and the SCSI bus. The MSB of the command byte differentiates DMA commands from non-DMA commands.

### Summary of Commands

| Command                          | Command Code (Hex.) |          |
|----------------------------------|---------------------|----------|
|                                  | Non-DMA Mode        | DMA Mode |
| <b>Initiator Commands</b>        |                     |          |
| Information Transfer             | 10                  | 90       |
| Initiator Command Complete Steps | 11                  | 91       |
| Message Accepted                 | 12                  | -        |
| Transfer Pad Bytes               | 18                  | 98       |
| Set ATN                          | 1A                  | -        |
| Reset ATN                        | 1B                  | -        |
| <b>Target Commands</b>           |                     |          |
| Send Message                     | 20                  | A0       |
| Send Status                      | 21                  | A1       |
| Send Data                        | 22                  | A2       |
| Disconnect Steps                 | 23                  | A3       |
| Terminate Steps                  | 24                  | A4       |
| Target Command Complete Steps    | 25                  | A5       |
| Disconnect                       | 27                  | A7       |
| Receive Message                  | 28                  | A8       |
| Receive Command Steps            | 29                  | A9       |
| Receive Data                     | 2A                  | AA       |
| Receive Command Steps            | 2B                  | AB       |
| Target Abort DMA                 | 04                  | 84       |

| Command  | Command Code (Hex.) |          |
|--|---------------------|----------|
|  | Non-DMA Mode        | DMA Mode |
| <b>Idle State Commands</b>                         |                     |          |
| Reselect Steps                                     | 40                  | C0       |
| Select without $\overline{\text{ATN}}$ Steps       | 41                  | C1       |
| Select with $\overline{\text{ATN}}$ Steps          | 42                  | C2       |
| Select with $\overline{\text{ATN}}$ and Stop Steps | 43                  | C3       |
| Enable Selection/Reselection                       | 44                  | C4       |
| Disable Selection/Reselection                      |                     | 45       |
| Select with $\overline{\text{ATN3}}$               | 46                  | C6       |
| <b>General Commands</b>                            |                     |          |
| No Operation                                       | 00                  | 80       |
| Clear FIFO   | 01                  | 81       |
| Reset Device                                       | 02                  | 82       |
| Reset SCSI bus                                     | 03                  | 83       |



**ABSOLUTE MAXIMUM RATINGS**

Storage Temperature ..... -55 to +125°C  
 Ambient Operating Temperature ..... 0 to +70°C  
 Maximum  $V_{CC}$  ..... -0.5 to +7.0 V  
 DC Voltage Applied to Any Pin . -0.5 to ( $V_{DD} + 0.3$ ) V  
 Input Static Discharge Protection . . 3000 V pin to pin  
 (Human body model: 100 pF at 1.5 K $\Omega$ )

*Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.*

**OPERATING RANGES****Commercial (C) Devices**

Ambient Temperature ( $T_A$ ) ..... 0 to +70°C  
 Supply Voltage ( $V_{DD}$ ) ..... 3.3 V  $\pm$  10%

*Operating ranges define those limits between which the functionality of the device is guaranteed.*

**DC OPERATING CHARACTERISTICS**  $V_{DD}=2.9$  V to 3.7 V;  $T_{CASE} = 0^\circ$  to + 100°

| Parameter Symbol          | Parameter Description   | Pin Names  | Test Condition                          | Min.     | Max.           | Unit          |
|---------------------------|-------------------------|--|---|----------|----------------|---------------|
| $I_{CCS}$                 | Static Supply Current   |  |   |          | 4.0            | mA            |
| $I_{CCD}$                 | Dynamic Supply Current  |  |   |          | 50             | mA            |
| $I_{LU}$                  | Latch Up Current        |  |   | -100     | +100           | mA            |
| <b>SCSI Pins</b>          |                         |  |   |          |                |               |
| $V_{IH}$                  | Input High Voltage      | All SCSI Inputs  |   | 2.0      | $V_{DD} + 0.3$ | V             |
| $V_{IL}$                  | Input Low Voltage       | All SCSI Inputs  |   | -0.3     | 0.8            | V             |
| $V_{IHST}$                | Input Hysteresis        | All SCSI Inputs  | $2.9\text{ V} < V_{DD} < 3.7\text{ V}$  | 200      |                | mV            |
| $V_{OH}$                  | Output High Voltage     | SD 7-0, SDP  |   |          |                |               |
| $V_{SOL1}$                | SCSI Output Low Voltage | SD 7-0, SDP  | $I_{OL} = 4\text{ mA}$                  | $V_{SS}$ | 0.4            | V             |
| $V_{SOL2}$                | SCSI Output Low Voltage | SDC 7-0, SDCP, MSG, C/D, I/O, ATN, RSTC, SELC, BSYC, ACKC and REQC | $I_{OL} = 48\text{ mA}$                 | $V_{SS}$ | 0.5            | V             |
| $I_{LI}$                  | Input Leakage           |  | $0.0\text{ V} \leq V_{IN} \leq V_{DD}$  | -10      | +10            | $\mu\text{A}$ |
| $I_{LO}$                  | Output Leakage          |  | $0.1\text{ V} \leq V_{OUT} \leq V_{DD}$ | -10      | +10            | $\mu\text{A}$ |
| <b>Bidirectional Pins</b> |                         |  |   |          |                |               |
| $V_{IH}$                  | Input High Voltage      |  |   | 2.0      | $V_{DD} + 0.3$ | V             |
| $V_{IL}$                  | Input Low Voltage       |  |   | -0.3     | 0.8            | V             |
| $V_{OH}$                  | Output High Voltage     | DMA 15-0, DMAP 1-0 and AD 7-0                                      |   |          |                |               |
| $V_{OL}$                  | Output Low Voltage      | DMA 15-0, DMAP 1-0 and AD 7-0                                      | $I_{OL} = 0.5\text{ mA}$                | $V_{SS}$ | 0.2            | V             |
|                           |                         |  | $I_{OL} = 2\text{ mA}$                  |          | 0.45           | V             |
| $I_{LI}$                  | Input Leakage           | DMA 15-0, DMAP 1-0 and AD 7-0                                      | $0.0\text{ V} \leq V_{IN} \leq V_{DD}$  |          | $\pm 15$       | $\mu\text{A}$ |
| $I_{LO}$                  | Output Leakage          | DMA 15-0, DMAP 1-0 and AD 7-0                                      | $0.1\text{ V} \leq V_{OUT} \leq V_{DD}$ |          | $\pm 15$       | $\mu\text{A}$ |
| <b>Output Pins</b>        |                         |  |   |          |                |               |
| $V_{OH}$                  | Output High Voltage     | DRQ and INT  |   |          |                |               |
| $V_{OL}$                  | Output Low Voltage      | DRQ and INT  | $I_{OL} = 0.5\text{ mA}$                |          | 0.2            | V             |
|                           |                         |  | $I_{OL} = 2.5\text{ mA}$                |          | 0.45           | V             |

2-30

# **Chapter 3**

## **One Time Programmable EPROM Products**

## CHAPTER 3

### One Time Programmable EPROM Products

|                            |      |
|----------------------------|------|
| Am27LV512 Data Sheet ..... | 3-3  |
| Am27LV010 Data Sheet ..... | 3-17 |
| Am27LV020 Data Sheet ..... | 3-33 |



# Am27LV512

**65,536 x 8-Bit CMOS Low Voltage, One Time Programmable Memory**

## DISTINCTIVE CHARACTERISTICS

- **3.3 V  $\pm$  0.3 V  $V_{CC}$  read operation**
- **High performance at 3.3 V  $V_{CC}$** 
  - 200 ns maximum access time
- **Low power consumption**
  - 90  $\mu$ W maximum standby power
  - 25  $\mu$ A maximum standby current
  - 54 mW maximum power at 5 MHz
  - 15 mA maximum current at 5 MHz
  - No data retention power
- **Industry standard packaging**
  - 32-pin PLCC
  - 32-pin Plastic DIP
- **Program voltage  $12.75 \pm 0.25$  V**
- **Latch-up protected to 100 mA from  $-1$  V to  $V_{CC} + 1$  V**
- **Flashrite™ programming**
  - 10  $\mu$ s typical byte-program
  - Less than 1 second typical chip program
- **Advanced CMOS memory technology**
  - Low cost single transistor memory cell

## GENERAL DESCRIPTION

The Am27LV512 device is a low voltage, low power, CMOS 64K x 8 One Time Programmable (OTP) non-volatile memory.

Maximum power consumption in standby mode is 90  $\mu$ W. If the device is constantly accessed at 5 MHz, then maximum power consumption increases to 54 mW. These power ratings are significantly lower than typical EPROM devices. Since power consumption is proportional to voltage squared, 3.3 V devices typically consume at least 57% less power than 5.0 V devices.

The Am27LV512 typically draws 10 mA of current enabling 200 ns read operations. Typical power consumption under these conditions equals 33 mW. This "high performance", low voltage device is ideal for BIOS storage in portable computing applications and control code storage in portable digital cellular phone applications.

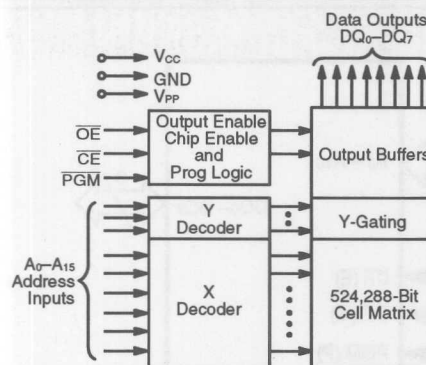
Low voltage CMOS designs require less operating power and hence dramatically increases the usable operating life of battery powered systems.

The Am27LV512 is packaged in standard 32-pin PLCC and Plastic DIP packages. It is designed to be programmed in standard EPROM programmers.

The highest degree of latch-up protection is achieved with AMD's proprietary non-epi process. Latch-up protection is provided for stresses up to 100 milliamps on address and data pins from  $-1$  V to  $V_{CC} + 1$  V.

The Am27LV512 is byte programmable using 10  $\mu$ s programming pulses in accordance with AMD's Flashrite programming algorithm. The typical room temperature programming time of the Am27LV512 is less than one second.

## BLOCK DIAGRAM



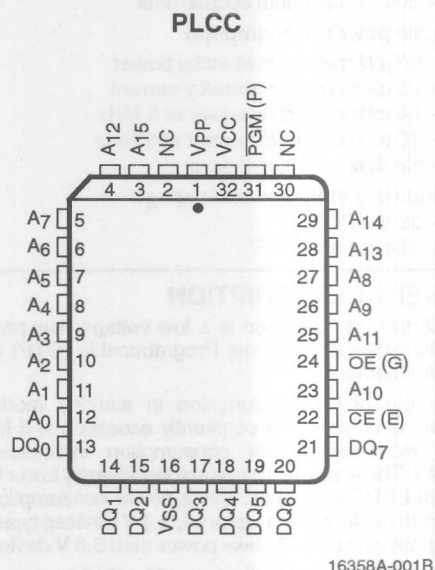
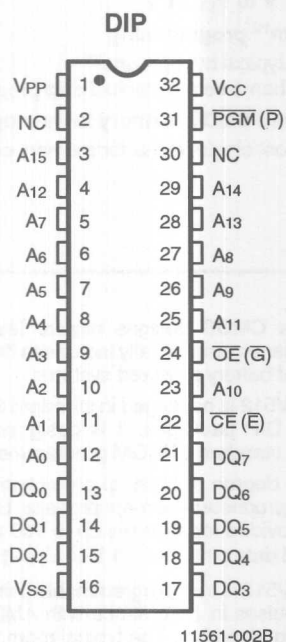
08140-001A

This document contains information on a product under development at Advanced Micro Devices, Inc. The information is intended to help you evaluate this product. AMD reserves the right to change or discontinue work on this proposed product without notice.

# PRODUCT SELECTOR GUIDE

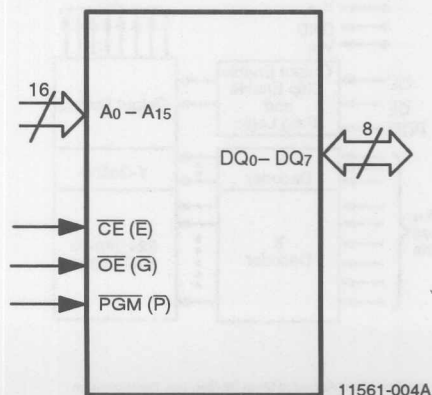
| Family Part No.                       | Am27LV512 |      |      |
|---------------------------------------|-----------|------|------|
| Ordering Part No:                     |           |      |      |
| $\pm 0.3 \text{ V } V_{CC}$ Tolerance | -200      | -250 | -300 |
| Max Access Time (ns)                  | 200       | 250  | 300  |
| $\overline{CE}$ (E) Access (ns)       | 200       | 250  | 300  |
| $\overline{OE}$ (G) Access (ns)       | 75        | 100  | 100  |

## CONNECTION DIAGRAMS



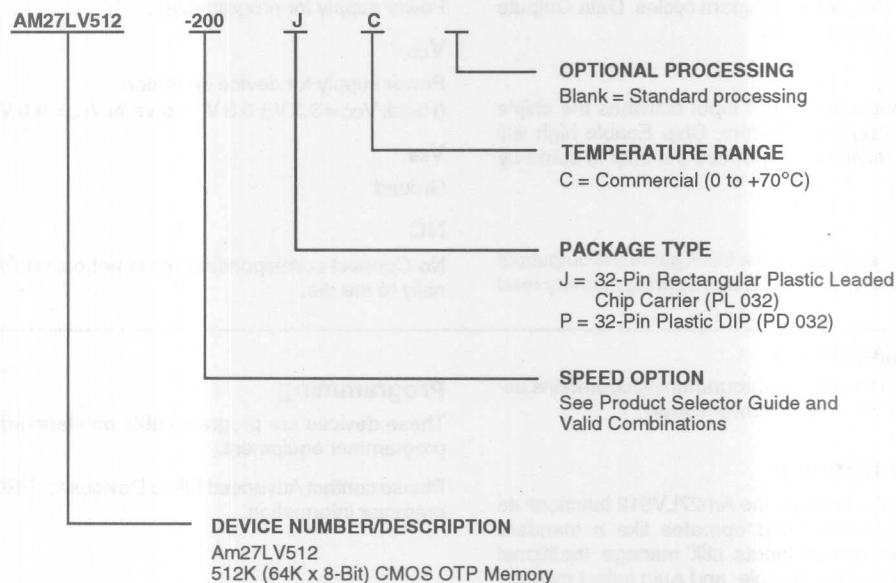
**Note:** Pin 1 is marked for orientation.

## LOGIC SYMBOL



**ORDERING INFORMATION****Standard Products**

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of these elements:



| Valid Combinations |        |
|--------------------|--------|
| Am27LV512-200      | JC, PC |
| Am27LV512-250      |        |
| Am27LV512-300      |        |

**Valid Combinations**

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

## PIN DESCRIPTION

**A<sub>0</sub> – A<sub>15</sub>**

Address Inputs for memory locations.

**DQ<sub>0</sub> – DQ<sub>7</sub>**

Data Inputs during memory program cycles. Internal latches hold data during program cycles. Data Outputs during memory read cycles.

**$\overline{\text{CE}}$  ( $\overline{\text{E}}$ )**

The Chip Enable active low input activates the chip's control logic and input buffers. Chip Enable high will deselect the device and operates the chip in stand-by mode.

**$\overline{\text{OE}}$  ( $\overline{\text{G}}$ )**

The Output Enable active low input gates the outputs of the device through the data buffers during memory read cycles.

**$\overline{\text{PGM}}$  ( $\overline{\text{P}}$ )**

The Program Enable active low input controls the program function of the memory array.

**V<sub>PP</sub>**

Power supply for programming.

**V<sub>CC</sub>**

Power supply for device operation.  
(Read: V<sub>CC</sub> = 3.3 V ± 0.3 V, Program: V<sub>CC</sub> = 5.0 V ± 10%)

**V<sub>SS</sub>**

Ground

**NC**

No Connect-corresponding pin is not connected internally to the die.

## BASIC PRINCIPLES

The Am27LV512 supports programming operations using a fixed 12.75 ± .25 V power supply.

### Read Only Memory

Without high V<sub>PP</sub> voltage, the Am27LV512 functions as a read only memory and operates like a standard EPROM. The control inputs still manage traditional read, standby, output disable, and Auto select modes.

### Programming

These devices are programmable on standard PROM programmer equipment.

Please contact Advanced Micro Devices for PROM programmer information.

## FUNCTIONAL DESCRIPTION

### Description Of User Modes

Table 1. Am27LV512 User Bus Operations

| Operation |                               | $\overline{\text{CE}}$<br>( $\overline{\text{E}}$ ) | $\overline{\text{OE}}$<br>( $\overline{\text{G}}$ ) | $\overline{\text{WE}}$<br>( $\overline{\text{W}}$ ) | V <sub>PP</sub><br>(Note 1) | A <sub>0</sub>  | A <sub>9</sub>              | I/O              |
|-----------|-------------------------------|---|---|---|-----------------------------|-----------------|-----------------------------|------------------|
| Read-Only | Read                          | V <sub>IL</sub>                                     | V <sub>IL</sub>                                     | X   | V <sub>PPL</sub>            | A <sub>0</sub>  | A <sub>9</sub>              | D <sub>OUT</sub> |
|           | Standby                       | V <sub>IH</sub>                                     | X   | X   | V <sub>PPL</sub>            | X               | X                           | HIGH Z           |
|           | Output Disable                | V <sub>IL</sub>                                     | V <sub>IH</sub>                                     | V <sub>IH</sub>                                     | V <sub>PPL</sub>            | X               | X                           | HIGH Z           |
|           | Auto-select Manufacturer Code | V <sub>IL</sub>                                     | V <sub>IL</sub>                                     | V <sub>IH</sub>                                     | V <sub>PPL</sub>            | V <sub>IL</sub> | V <sub>ID</sub><br>(Note 2) | CODE<br>(01H)    |
|           | Auto-select Device Code       | V <sub>IL</sub>                                     | V <sub>IL</sub>                                     | V <sub>IH</sub>                                     | V <sub>PPL</sub>            | V <sub>IH</sub> | V <sub>ID</sub><br>(Note 2) | CODE<br>(26H)    |

#### Legend:

X = Don't care, where Don't Care is either V<sub>IL</sub> or V<sub>IH</sub> levels, V<sub>PPL</sub> = V<sub>PP</sub> < V<sub>CC</sub> + 2 V, See DC Characteristics for voltage levels of V<sub>PPH</sub>, 0 V < A<sub>n</sub> < V<sub>CC</sub> + 2 V, (normal CMOS input levels, where n = 0 or 9).

#### Notes:

- V<sub>PPL</sub> may be grounded, connected with a resistor to ground, or ≤ V<sub>CC</sub> + 2.0V. V<sub>PPH</sub> is the programming voltage specified for the device. Refer to the DC characteristics. When V<sub>PP</sub> = V<sub>PPL</sub>, memory contents can be read but not written.
- 11.5 ≤ V<sub>ID</sub> ≤ 13.0 V, V<sub>CC</sub> = 5.0 V ± 10%.



**READ ONLY MODE** **$V_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$** **Read**

The Am27LV512 functions as a read only memory. The Am27LV512 has two control functions. Both must be satisfied in order to output data.  $\overline{CE}$  controls power to the device. This pin should be used for specific device selection.  $\overline{OE}$  controls the device outputs and should be used to gate data to the output pins if a device is selected.

Address access time  $t_{ACC}$  is equal to the delay from stable addresses to valid output data. The chip enable access time  $t_{CE}$  is the delay from stable addresses and stable  $\overline{CE}$  to valid data at the output pins. The output enable access time is the delay from the falling edge of  $\overline{OE}$  to valid data at the output pins (assuming the addresses have been stable at least  $t_{ACC} - t_{OE}$ ).

**Standby Mode**

The Am27LV512 has one standby mode. The CMOS standby mode ( $\overline{CE}$  input held at  $V_{CC} \pm 0.5\text{ V}$ ), consumes less than  $25\text{ }\mu\text{A}$  of current. When in the standby mode the outputs are in a high impedance state, independent of the  $\overline{OE}$  input.

If the device is deselected during programming, or program verification, the device will draw active current until the operation is terminated.

**Output Disable**

Output from the device is disabled when  $\overline{OE}$  is at a logic high level. When disabled, output pins are in a high impedance state.

**Auto Select**

The Am27LV512 can be programmed in a standard PROM programmer.

The Auto select mode allows the reading out of a binary code from the device that will identify its manufacturer and type. This mode is intended for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm.

**Programming In A PROM Programmer**

To activate this mode, the programming equipment must force  $V_{ID}$  (11.5 V to 13.0 V) on address  $A_9$ . Two identifier bytes may then be sequenced from the device outputs by toggling address  $A_0$  from  $V_{IL}$  to  $V_{IH}$ . All other address lines must be held at  $V_{IL}$ , and  $V_{PP}$  must be less than or equal to  $V_{CC} + 2.0\text{ V}$  while using this Auto select mode. Byte 0 ( $A_0 = V_{IL}$ ) represents the manufacturer code and byte 1 ( $A_0 = V_{IH}$ ) the device identifier code. For the Am27LV512 these two bytes are given in the table below. All identifiers for manufacturer and device codes will exhibit odd parity with the MSB ( $DQ_7$ ) defined as the parity bit.

**Table 2. Am27LV512 Auto Select Code**

| Type              | $A_0$    | Code (HEX) | $DQ_7$ | $DQ_6$ | $DQ_5$ | $DQ_4$ | $DQ_3$ | $DQ_2$ | $DQ_1$ | $DQ_0$ |
|-------------------|----------|------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Manufacturer Code | $V_{IL}$ | 01         | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 1      |
| Device Code       | $V_{IH}$ | 26         | 0      | 0      | 1      | 0      | 0      | 1      | 1      | 0      |

## ABSOLUTE MAXIMUM RATINGS

|  |                   |
|--|-------------------|
| Storage Temperature  |                   |
| Plastic Packages   | – 65°C to +125°C  |
| Ambient Temperature  |                   |
| with Power Applied   | – 55°C to + 125°C |
| Voltage with Respect To Ground All pins except A <sub>9</sub> and V <sub>PP</sub> (Note 1) | – 2.0 V to 7.0 V  |
| V <sub>CC</sub> (Note 1)   | – 2.0 V to 7.0 V  |
| A <sub>9</sub> (Note 2)  | – 2.0 V to 14.0 V |
| V <sub>PP</sub> (Note 2)   | – 2.0 V to 14.0 V |
| Output Short Circuit Current (Note 3)  | 200 mA            |

### Notes:

1. Minimum DC voltage on input or I/O pins is –0.5 V. During voltage transitions, inputs may overshoot V<sub>SS</sub> to –2.0 V for periods of up to 20 ns. Maximum DC voltage on output and I/O pins is V<sub>CC</sub> + 0.5 V. During voltage transitions, outputs may overshoot to V<sub>CC</sub> + 2.0 V for periods up to 20 ns.
2. Minimum DC input voltage on A<sub>9</sub> and V<sub>PP</sub> pins is –0.5V. During voltage transitions, A<sub>9</sub> and V<sub>PP</sub> may overshoot V<sub>SS</sub> to –2.0 V for periods of up to 20 ns. Maximum DC input voltage on A<sub>9</sub> and V<sub>PP</sub> is +13.5 V which may overshoot to 14.0 V for periods up to 20 ns.
3. No more than one output shorted at a time. Duration of the short circuit should not be greater than one second.

Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure of the device to absolute maximum rating conditions for extended periods may affect device reliability.

## OPERATING RANGES

### Commercial (C) Devices

|                                    |              |
|------------------------------------|--------------|
| Case Temperature (T <sub>C</sub> ) | 0°C to +70°C |
|------------------------------------|--------------|

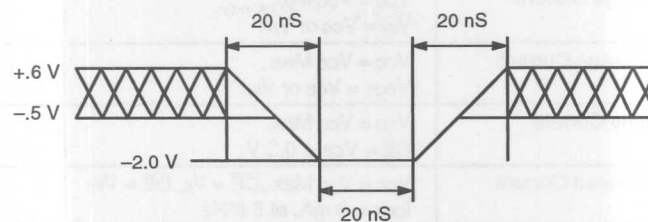
### V<sub>CC</sub> Supply Voltage

|                               |                  |
|-------------------------------|------------------|
| V <sub>CC</sub> for Am27LV512 | +3.0 V to +3.6 V |
|-------------------------------|------------------|

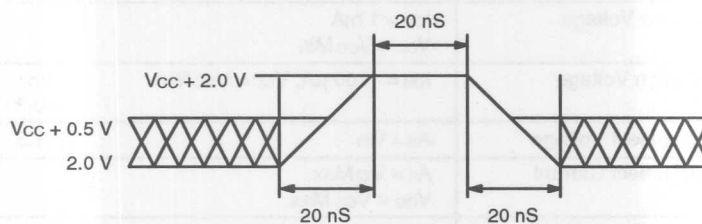
### V<sub>PP</sub> Supply Voltage

|                    |                  |
|--------------------|------------------|
| Program and Verify | +12.5 V to +13 V |
|--------------------|------------------|

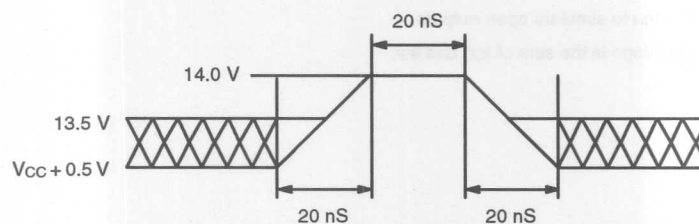
Operating ranges define those limits between which the functionality of the device is guaranteed.

**MAXIMUM OVERSHOOT****Maximum Negative Input Overshoot**

11561-009B

**Maximum Negative Overshoot Waveform****Maximum Positive Input Overshoot**

11561-010A

**Maximum Positive Overshoot Waveform****Maximum  $V_{PP}$  Overshoot**

11561-011A

**Maximum  $V_{PP}$  Overshoot Waveform**

## DC CHARACTERISTICS-CMOS COMPATIBLE

| Parameter Symbol | Parameter Description                 | Test Conditions   | Min.           | Max.           | Unit    |
|------------------|---------------------------------------|---|----------------|----------------|---------|
| $I_{LI}$         | Input Leakage Current                 | $V_{CC} = V_{CC \text{ Max.}}$ ,<br>$V_{IN} = V_{CC}$ or $V_{SS}$   |                | + 1.0          | $\mu A$ |
| $I_{LO}$         | Output Leakage Current                | $V_{CC} = V_{CC \text{ Max.}}$ ,<br>$V_{OUT} = V_{CC}$ or $V_{SS}$  |                | + 1.0          | $\mu A$ |
| $I_{CCS}$        | $V_{CC}$ Standby Current              | $V_{CC} = V_{CC \text{ Max.}}$ ,<br>$\overline{CE} = V_{CC} \pm 0.3 \text{ V}$  |                | 25             | $\mu A$ |
| $I_{CC1}$        | $V_{CC}$ Active Read Current          | $V_{CC} = V_{CC \text{ Max.}}$ , $\overline{CE} = V_{IL}$ , $\overline{OE} = V_{IH}$<br>$I_{OUT} = 0 \text{ mA}$ , at 5 MHz |                | 15             | mA      |
| $I_{CC2}$        | $V_{CC}$ Programming Current          | $\overline{CE} = V_{IL}$<br>Programming in Progress   |                | 30             | mA      |
| $I_{PPS}$        | $V_{PP}$ Standby Current              | $V_{PP} = V_{PPL}$  |                | + 1.0          | $\mu A$ |
| $V_{IL}$         | Input Low Voltage                     |   | -0.5           | 0.6            | V       |
| $V_{IH}$         | Input High Voltage                    |   | 2.0            | $V_{CC} + 0.5$ | V       |
| $V_{OL}$         | Output Low Voltage                    | $I_{OL} = 1 \text{ mA}$<br>$V_{CC} = V_{CC \text{ Min.}}$   |                | 0.3            | V       |
| $V_{OH}$         | Output High Voltage                   | $I_{OH} = -100 \mu A$ , $V_{CC} = V_{CC \text{ Min.}}$  | $V_{CC} - 0.3$ |                | V       |
| $V_{ID}$         | $A_9$ Auto Select Voltage             | $A_9 = V_{ID}$  | 11.5           | 13.0           | V       |
| $I_{ID}$         | $A_9$ Auto Select Current             | $A_9 = V_{ID \text{ Max.}}$ ,<br>$V_{CC} = V_{CC \text{ Max.}}$   |                | 35             | $\mu A$ |
| $V_{PPL}$        | $V_{PP}$ during Read-Only Operations  |   | 0.0            | $V_{CC} + 2.0$ | V       |
| $V_{PPH}$        | $V_{PP}$ during Read/Write Operations |   | 12.5           | 13.0           | V       |

## Notes:

1. **Caution:** the Am27LV512 must not be removed from (or inserted into) a socket when  $V_{CC}$  or  $V_{PP}$  is applied.
2.  $I_{CC1}$  is tested with  $\overline{OE} = V_{IH}$  to simulate open outputs.
3. Maximum active power usage is the sum of  $I_{CC}$  and  $I_{PP}$ .

**PIN CAPACITANCE**

| Parameter Symbol | Parameter Description             | Test Conditions      | Typ. | Max. | Unit |
|------------------|-----------------------------------|----------------------|------|------|------|
| C <sub>IN</sub>  | Input Capacitance                 | V <sub>IN</sub> = 0  | 8    | 10   | pF   |
| C <sub>OUT</sub> | Output Capacitance                | V <sub>OUT</sub> = 0 | 8    | 12   | pF   |
| C <sub>IN2</sub> | V <sub>PP</sub> Input Capacitance | V <sub>PP</sub> = 0  | 8    | 12   | pF   |

**Notes:**

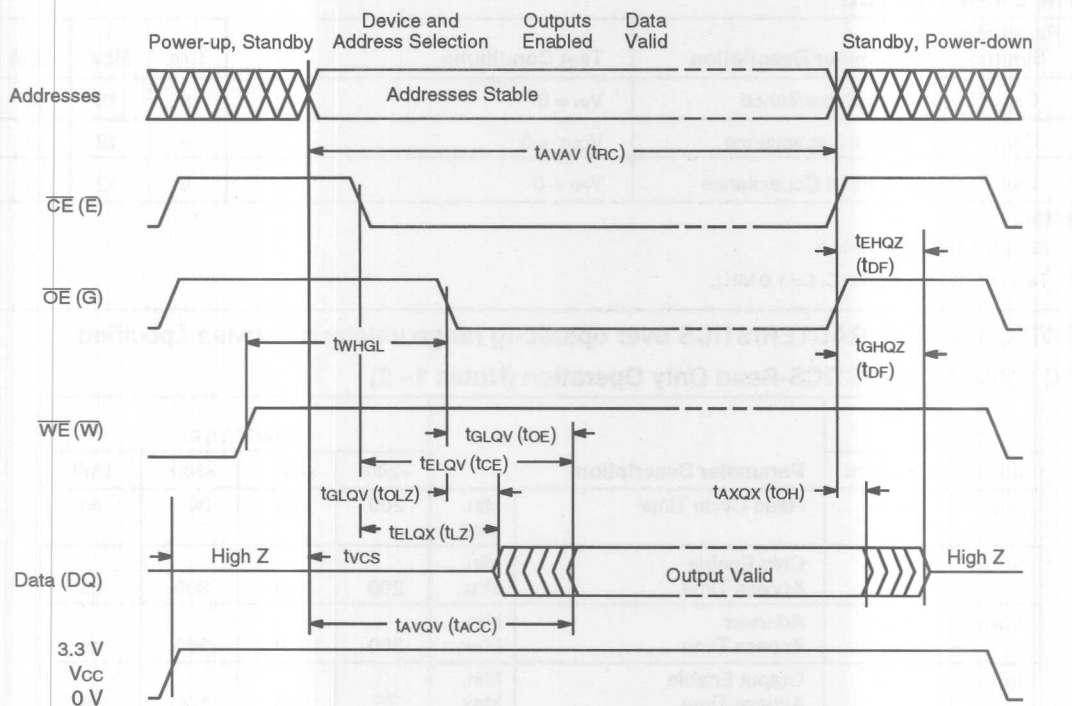
1. Sampled, not 100% tested.
2. Test conditions T<sub>A</sub> = 25°C, f = 1.0 MHz.

**SWITCHING CHARACTERISTICS over operating range unless otherwise specified****AC CHARACTERISTICS-Read Only Operation (Notes 1– 2)**

| Parameter Symbols |                  | Parameter Description                               |              | Am27LV512 |      |      |      |
|-------------------|------------------|---|--------------|-----------|------|------|------|
| JEDEC             | Standard         |   |              | -200      | -250 | -300 | Unit |
| t <sub>AVAV</sub> | t <sub>RC</sub>  | Read Cycle Time                                     | Min.<br>Max. | 200       | 250  | 300  | ns   |
| t <sub>ELQV</sub> | t <sub>CE</sub>  | Chip Enable Access Time                             | Min.<br>Max. | 200       | 250  | 300  | ns   |
| t <sub>AVQV</sub> | t <sub>ACC</sub> | Address Access Time                                 | Min.<br>Max. | 200       | 250  | 300  | ns   |
| t <sub>GLQV</sub> | t <sub>OE</sub>  | Output Enable Access Time                           | Min.<br>Max. | 75        | 100  | 100  | ns   |
| t <sub>ELQX</sub> | t <sub>LZ</sub>  | Chip Enable to Output in Low Z                      | Min.<br>Max. | 0         | 0    | 0    | ns   |
| t <sub>EHQZ</sub> | t <sub>DF</sub>  | Chip Disable to Output in High Z                    | Min.<br>Max. | 35        | 35   | 35   | ns   |
| t <sub>GLQX</sub> | t <sub>OLZ</sub> | Output Enable to Output in Low Z                    | Min.<br>Max. | 0         | 0    | 0    | ns   |
| t <sub>GHQZ</sub> | t <sub>DF</sub>  | Output Disable to Output in High Z                  | Min.<br>Max. | 35        | 35   | 35   | ns   |
| t <sub>AXQX</sub> | t <sub>OH</sub>  | Output Hold from first of Address, CE, or OE Change | Min.<br>Max. | 0         | 0    | 0    | ns   |
| t <sub>VCS</sub>  |                  | V <sub>CC</sub> Set-up Time to Valid Read           | Min.<br>Max. | 50        | 50   | 50   | μs   |

**Notes:**

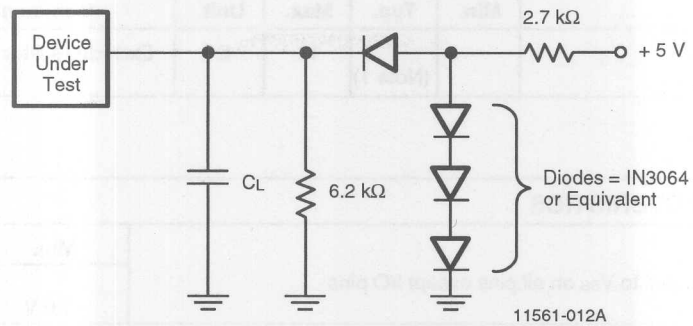
1. Output Load: 1 TTL gate and C<sub>L</sub> = 100 pF, Input Rise and Fall Times: ≤ 10 ns, Input Pulse levels: 0.45 to 2.4 V, Timing Measurement Reference Level - Inputs: 0.6 V and 2 V  
Outputs: 1.5 V
2. t<sub>VCS</sub> is guaranteed by design not tested.



11561-013C

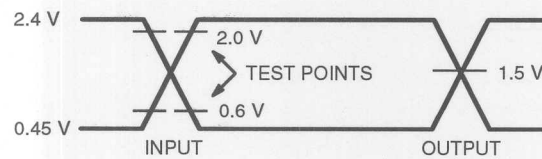
AC Waveforms for Read Operations

## SWITCHING TEST CIRCUIT



$C_L = 100$  pF including jig capacitance

## SWITCHING TEST WAVEFORMS



All Devices

AC Testing: Inputs are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0". Input pulse rise and fall times are  $\leq 10$  ns.

16357A-002B

**ERASE AND PROGRAMMING PERFORMANCE**

| Parameter             | Limits |               |      | Unit | Comments                       |
|-----------------------|--------|---------------|------|------|--------------------------------|
|                       | Min.   | Typ.          | Max. |      |                                |
| Chip Programming Time |        | 1<br>(Note 1) | 12   | S    | Excludes system-level overhead |

**Note:**

1. 25°C, 12.75 V V<sub>PP</sub>

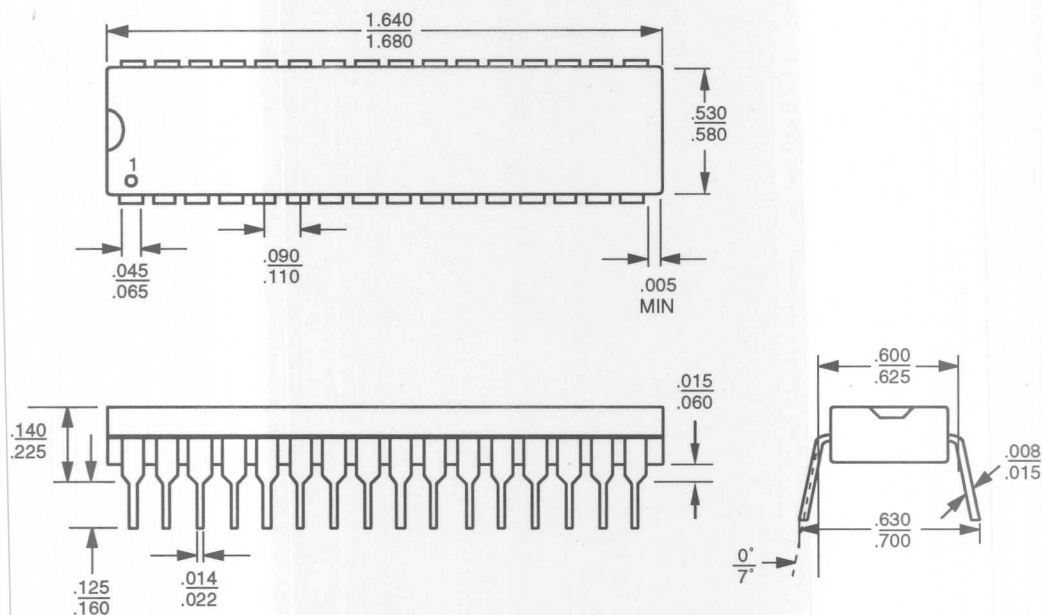
**LATCHUP CHARACTERISTICS**

|  | Min.    | Max.                    |
|--|---------|-------------------------|
| Input Voltage with respect to V <sub>SS</sub> on all pins except I/O pins<br>(Including A <sub>9</sub> and V <sub>PP</sub> ) | -1.0 V  | 13.5 V                  |
| Input Voltage with respect to V <sub>SS</sub> on all pins I/O pins   | -1.0 V  | V <sub>CC</sub> + 1.0 V |
| Current  | -100 mA | +100 mA                 |
| Includes all pins except V <sub>CC</sub> . Test conditions: V <sub>CC</sub> = 5.0 V, one pin at a time.                      |         |                         |



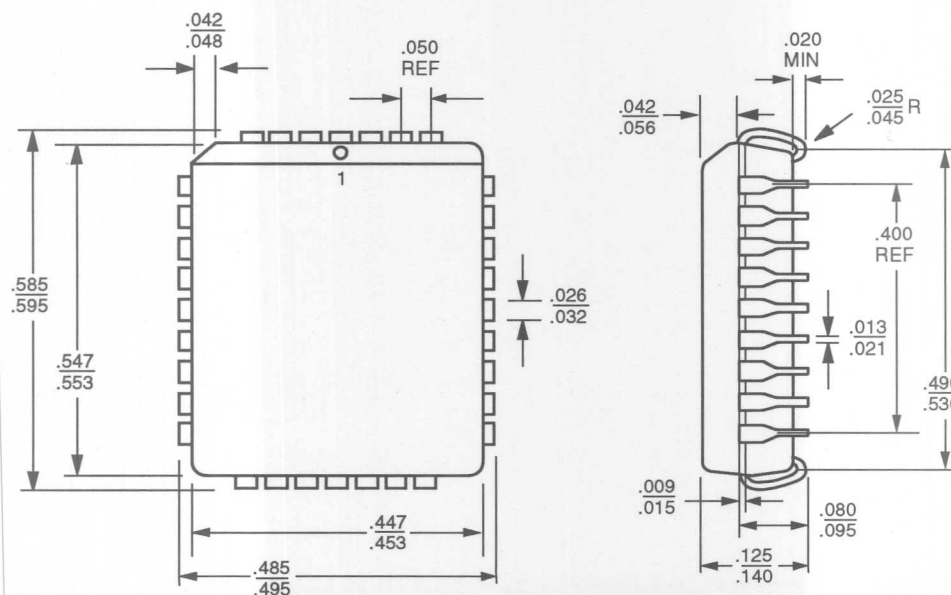
## PHYSICAL DIMENSIONS\*

## PD 032



12416B

## PL 032



06971C

\*For reference only. All dimensions are measured in inches, unless otherwise noted. BSC is an ANSI standard for Basic Space Centering.





# Am27LV010

## 131,072 x 8-Bit CMOS Low Voltage, One Time Programmable Memory

### DISTINCTIVE CHARACTERISTICS

- **3.3 V  $\pm$  0.3 V  $V_{CC}$  read operation**
- **High performance at 3.3 V $_{CC}$** 
  - 200 ns maximum access time
- **Low power consumption**
  - 90  $\mu$ W maximum standby power
  - 25  $\mu$ A maximum standby current
  - 54 mW maximum power at 5 MHz
  - 15 mA maximum current at 5 MHz
  - No data retention power
- **Industry standard packaging**
  - 32-pin PLCC
  - 32-pin Thin Small Outline Package
  - 32-pin Plastic DIP
- **Program voltage 12.75  $\pm$  0.25 V**
- **Latch-up protected to 100 mA from  $-1$  V to  $V_{CC} + 1$  V**
- **Flashrite™ programming**
  - 10  $\mu$ s typical byte-program
  - Less than 3 seconds typical chip program
- **Advanced CMOS memory technology**
  - Low cost single transistor memory cell

### GENERAL DESCRIPTION

The Am27LV010 device is a low voltage, low power, CMOS 128K x 8 One Time Programmable (OTP) non-volatile memory.

Maximum power consumption in standby mode is 90  $\mu$ W. If the device is constantly accessed at 5 MHz, then maximum power consumption increases to 54 mW. These power ratings are significantly lower than typical EPROM devices. Since power consumption is proportional to voltage squared, 3.3 V devices typically consume at least 57% less power than 5.0 V devices.

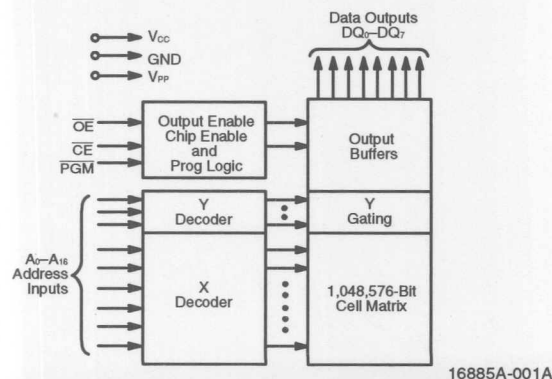
The Am27LV010 typically draws 10 mA of current enabling 200 ns read operations. Typical power consumption under these conditions equals 33 mW. This "high

performance", low voltage device is ideal for BIOS storage in portable computing applications and control code storage in portable digital cellular phone applications. Low voltage CMOS designs require less operating power and hence dramatically increases the usable operating life of battery powered systems.

The Am27LV010 is packaged in a 32-pin PLCC, Plastic DIP and Thin Small Outline Package (TSOP) versions. It is designed to be programmed in standard EPROM programmers.

The highest degree of latch-up protection is achieved with AMD's proprietary non-epi process. Latch-up protection is provided for stresses up to 100 milliamps on address and data pins from  $-1$  V to  $V_{CC} + 1$  V.

### BLOCK DIAGRAM

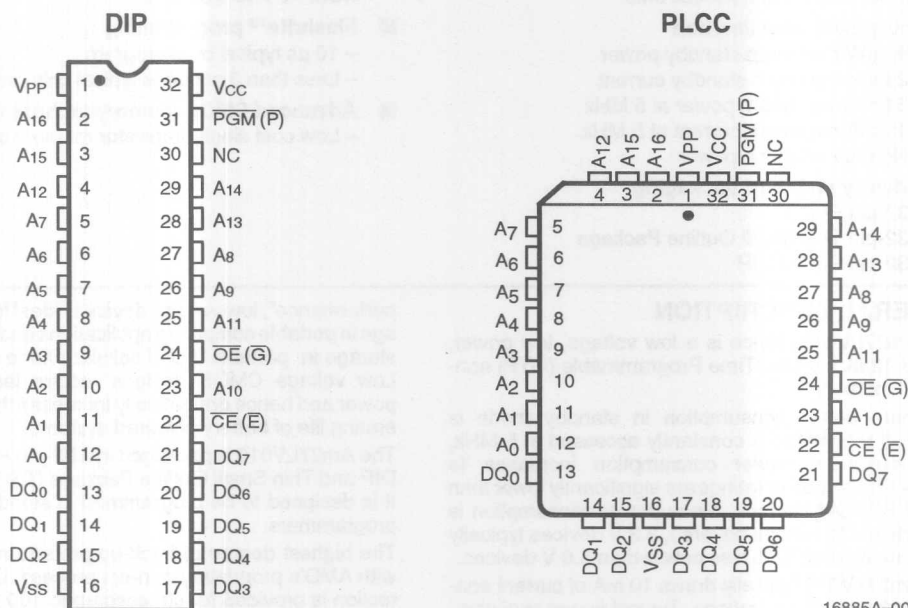


This document contains information on a product under development at Advanced Micro Devices, Inc. The information is intended to help you evaluate this product. AMD reserves the right to change or discontinue work on this proposed product without notice.

PRODUCT SELECTOR GUIDE

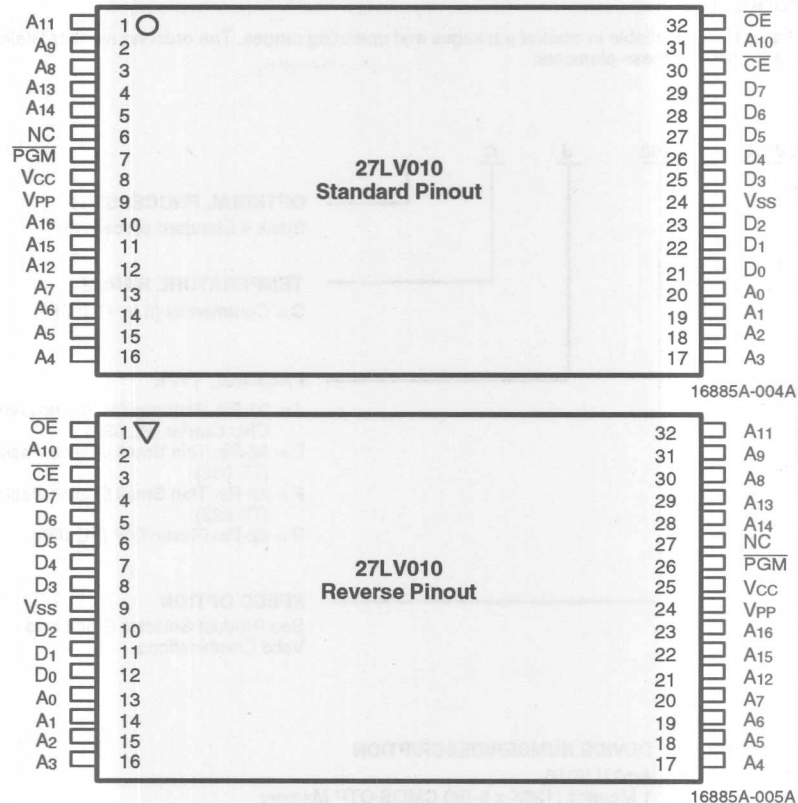
| Family Part No.                 | Am27LV010 |      |      |
|---------------------------------|-----------|------|------|
| Ordering Part No:               |           |      |      |
| $\pm 0.3$ V $V_{CC}$ Tolerance  | -200      | -250 | -300 |
| Max Access Time (ns)            | 200       | 250  | 300  |
| $\overline{CE}$ (E) Access (ns) | 200       | 250  | 300  |
| $\overline{OE}$ (G) Access (ns) | 75        | 100  | 100  |

CONNECTION DIAGRAMS



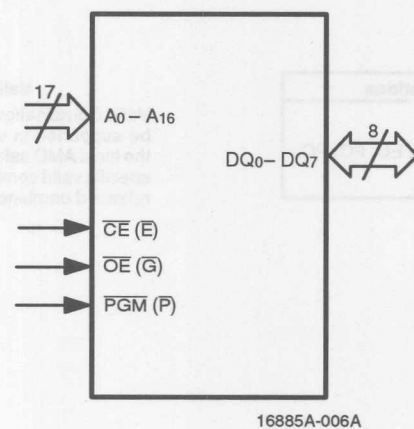
**Note:** Pin 1 is marked for orientation.

## TSOP PACKAGES



27LV020 256K x 8 OTP in 32 Lead TSOP

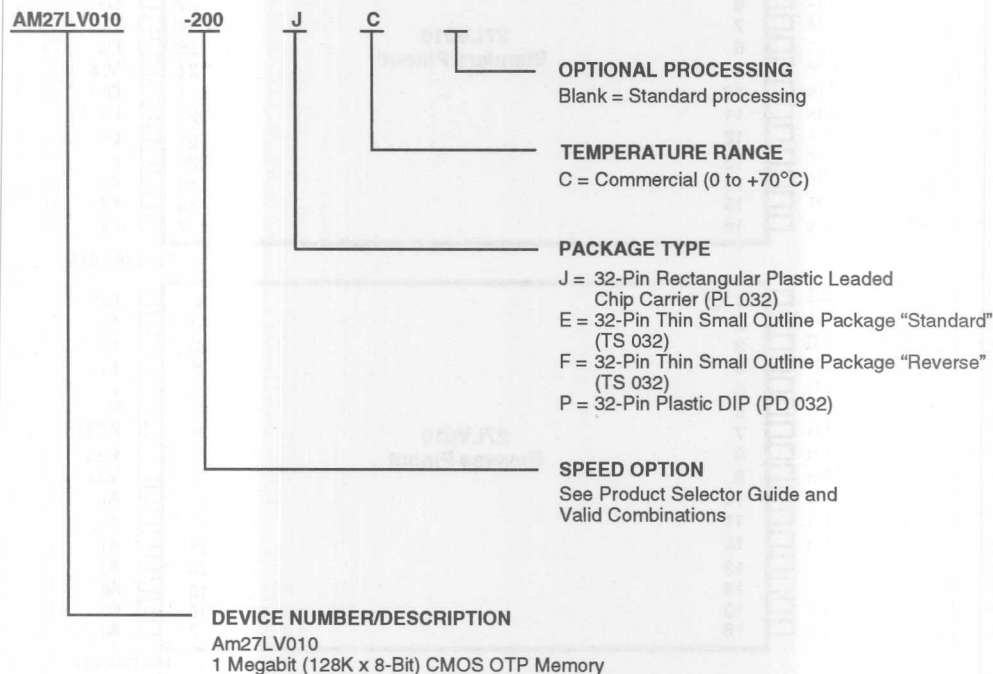
## LOGIC SYMBOL



## ORDERING INFORMATION

### Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of these elements:



| Valid Combinations |                |
|--------------------|----------------|
| Am27LV010-200      | JC, EC, FC, PC |
| Am27LV010-250      |                |
| Am27LV010-300      |                |

#### Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

**PIN DESCRIPTION****V<sub>PP</sub>**

Power supply for programming.

**V<sub>CC</sub>**

Power supply for device operation. (Read:  $V_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ , Program:  $V_{CC} = 5.0\text{ V} \pm 10\%$ .)

**V<sub>SS</sub>**

Ground

**NC**

No Connect-corresponding pin is not connected internally to the die.

**A<sub>0</sub> – A<sub>16</sub>**

Address Inputs for memory locations.

**DQ<sub>0</sub> – DQ<sub>7</sub>**

Data Inputs during memory program cycles. Internal latches hold data during program cycles. Data Outputs during memory read cycles.

 **$\overline{\text{CE}}$  ( $\overline{\text{E}}$ )**

The Chip Enable active low input activates the chip's control logic and input buffers. Chip Enable high will deselect the device and operates the chip in stand-by mode.

 **$\overline{\text{OE}}$  ( $\overline{\text{G}}$ )**

The Output Enable active low input gates the outputs of the device through the data buffers during memory read cycles.

 **$\overline{\text{PGM}}$  ( $\overline{\text{P}}$ )**

The Program Enable active low input controls the program function of the memory array.

**BASIC PRINCIPLES**

The Am27LV010 supports programming operations using a fixed  $12.75 \pm 0.25\text{ V}$  power supply.

**Read Only Memory**

Without high  $V_{PP}$  voltage, the Am27LV010 functions as a read only memory and operates like a standard EPROM. The control inputs still manage traditional read, standby, output disable, and Auto select modes.

**Programming**

These devices are programmable on standard PROM programmer equipment.

Please contact Advanced Micro Devices for PROM programmer information.

**FUNCTIONAL DESCRIPTION****Description Of User Modes****Table 1. Am27LV010 User Bus Operations**

| Operation |                               | $\overline{\text{CE}}$ ( $\overline{\text{E}}$ ) | $\overline{\text{OE}}$ ( $\overline{\text{G}}$ ) | $\overline{\text{PGM}}$ | V <sub>PP</sub><br>(Note 1) | A <sub>0</sub>  | A <sub>9</sub>              | I/O              |
|-----------|-------------------------------|--|--|-------------------------|-----------------------------|-----------------|-----------------------------|------------------|
| Read-Only | Read                          | V <sub>IL</sub>                                  | V <sub>IL</sub>                                  | X                       | V <sub>PPL</sub>            | A <sub>0</sub>  | A <sub>9</sub>              | D <sub>OUT</sub> |
|           | Standby                       | V <sub>IH</sub>                                  | X  | X                       | V <sub>PPL</sub>            | X               | X                           | HIGH Z           |
|           | Output Disable                | V <sub>IL</sub>                                  | V <sub>IH</sub>                                  | V <sub>IH</sub>         | V <sub>PPL</sub>            | X               | X                           | HIGH Z           |
|           | Auto-select Manufacturer Code | V <sub>IL</sub>                                  | V <sub>IL</sub>                                  | V <sub>IH</sub>         | V <sub>PPL</sub>            | V <sub>IL</sub> | V <sub>ID</sub><br>(Note 2) | CODE<br>(01H)    |
|           | Auto-select Device Code       | V <sub>IL</sub>                                  | V <sub>IL</sub>                                  | V <sub>IH</sub>         | V <sub>PPL</sub>            | V <sub>IH</sub> | V <sub>ID</sub><br>(Note 2) | CODE<br>(A8)     |

**Legend:**

X = Don't care, where Don't Care is either V<sub>IL</sub> or V<sub>IH</sub> levels, V<sub>PPL</sub> =  $V_{PP} < V_{CC} + 2\text{ V}$ , See DC Characteristics for voltage levels of V<sub>PPH</sub>,  $0\text{ V} < A_n < V_{CC} + 2\text{ V}$ , (normal CMOS input levels, where n = 0 or 9).

**Notes:**

1. V<sub>PPL</sub> may be grounded, connected with a resistor to ground, or  $\leq V_{CC} + 2.0\text{ V}$ . V<sub>PPH</sub> is the programming voltage specified for the device. Refer to the DC characteristics. When  $V_{PP} = V_{PPL}$ , memory contents can be read but not written.
2.  $11.5 \leq V_{ID} \leq 13.0\text{ V}$ ,  $V_{CC} = 5.0\text{ V} \pm 10\%$

## READ ONLY MODE

$V_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$

### Read

The Am27LV010 functions as a read only memory. The Am27LV010 has two control functions. Both must be satisfied in order to output data.  $\overline{CE}$  controls power to the device. This pin should be used for specific device selection.  $\overline{OE}$  controls the device outputs and should be used to gate data to the output pins if a device is selected.

Address access time  $t_{ACC}$  is equal to the delay from stable addresses to valid output data. The chip enable access time  $t_{CE}$  is the delay from stable addresses and stable  $\overline{CE}$  to valid data at the output pins. The output enable access time is the delay from the falling edge of  $\overline{OE}$  to valid data at the output pins (assuming the addresses have been stable at least  $t_{ACC} - t_{OE}$ ).

### Standby Mode

The Am27LV010 has one standby mode. The CMOS standby mode ( $\overline{CE}$  input held at  $V_{CC} \pm 0.5\text{V}$ ), consumes less than 25  $\mu\text{A}$  of current. When in the standby mode the outputs are in a high impedance state, independent of the  $\overline{OE}$  input.

If the device is deselected during programming, or program verification, the device will draw active current until the operation is terminated.

### Output Disable

Output from the device is disabled when  $\overline{OE}$  is at a logic high level. When disabled, output pins are in a high impedance state.

### Auto Select

The Am27LV010 can be programmed in a standard PROM programmer.

The Auto select mode allows the reading out of a binary code from the device that will identify its manufacturer and type. This mode is intended for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm.

### Programming In A PROM Programmer

To activate this mode, the programming equipment must force  $V_{ID}$  (11.5V to 13.0V) on address  $A_9$ . Two identifier bytes may then be sequenced from the device outputs by toggling address  $A_0$  from  $V_{IL}$  to  $V_{IH}$ . All other address lines must be held at  $V_{IL}$ , and  $V_{PP}$  must be less than or equal to  $V_{CC} + 2.0\text{V}$  while using this Auto select mode. Byte 0 ( $A_0 = V_{IL}$ ) represents the manufacturer code and byte 1 ( $A_0 = V_{IH}$ ) the device identifier code. For the Am27LV010 these two bytes are given in the table below. All identifiers for manufacturer and device codes will exhibit odd parity with the MSB ( $DQ_7$ ) defined as the parity bit.

Table 2. Am27LV020 Auto Select Code

| Type              | $A_0$    | Code (HEX) | $DQ_7$ | $DQ_6$ | $DQ_5$ | $DQ_4$ | $DQ_3$ | $DQ_2$ | $DQ_1$ | $DQ_0$ |
|-------------------|----------|------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Manufacturer Code | $V_{IL}$ | 01         | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 1      |
| Device Code       | $V_{IH}$ | A8         | 1      | 0      | 1      | 0      | 1      | 0      | 0      | 0      |



**ABSOLUTE MAXIMUM RATINGS**

|  |                   |
|--|-------------------|
| Storage Temperature                                |                   |
| Plastic Packages                                   | −65°C to +125°C   |
| Ambient Temperature                                |                   |
| with Power Applied                                 | −55°C to +125°C   |
| Voltage with Respect To Ground. All pins           |                   |
| except A <sub>9</sub> and V <sub>PP</sub> (Note 1) | −2.0 V to +7.0 V  |
| V <sub>CC</sub> (Note 1)                           | −2.0 V to +7.0 V  |
| A <sub>9</sub> (Note 2)                            | −2.0 V to +14.0 V |
| V <sub>PP</sub> (Note 2)                           | −2.0 V to +14.0 V |
| Output Short Circuit Current (Note 3)              | 200 mA            |

**Notes:**

1. Minimum DC voltage on input or I/O pins is −0.5 V. During voltage transitions, inputs may overshoot V<sub>SS</sub> to −2.0 V for periods of up to 20 ns. Maximum DC voltage on output and I/O pins is V<sub>CC</sub> + 0.5 V. During voltage transitions, outputs may overshoot to V<sub>CC</sub> + 2.0 V for periods up to 20 ns.
2. During programming operations only. Minimum DC input voltage on A<sub>9</sub> and V<sub>PP</sub> pins is −0.5V. During voltage transitions, A<sub>9</sub> and V<sub>PP</sub> may overshoot V<sub>SS</sub> to −2.0 V for periods of up to 20 ns. Maximum DC input voltage on A<sub>9</sub> and V<sub>PP</sub> is +13.5 V which may overshoot to 14.0 V for periods up to 20 ns.
3. No more than one output shorted at a time. Duration of the short circuit should not be greater than one second.

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure of the device to absolute maximum rating conditions for extended periods may affect device reliability.

**OPERATING RANGES****Commercial (C) Devices**

Case Temperature (T<sub>C</sub>) . . . . . 0°C to +70°C

**V<sub>CC</sub> Supply Voltages**

V<sub>CC</sub> for Am27LV010 . . . . . +3.0 V to +3.6 V

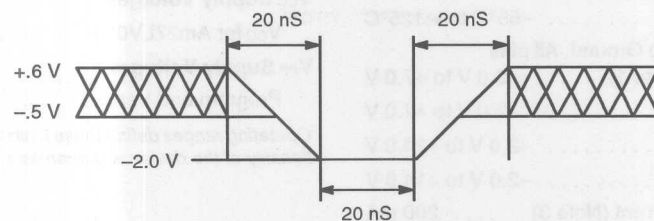
**V<sub>PP</sub> Supply Voltages**

Program and Verify . . . . . +12.5 V to +13 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

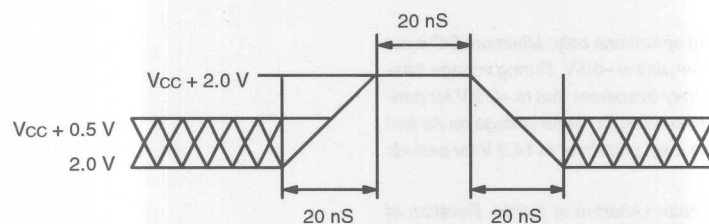
## MAXIMUM OVERSHOOT

### Maximum Negative Input Overshoot



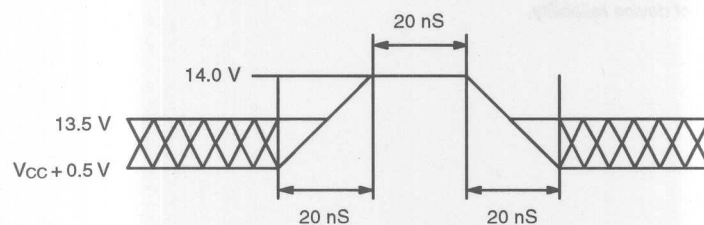
11561-009B

### Maximum Positive Input Overshoot



11561-010A

### Maximum $V_{PP}$ Overshoot



11561-011A

## DC CHARACTERISTICS-CMOS COMPATIBLE

| Parameter Symbol | Parameter Description                 | Test Conditions   | Min.           | Max.           | Unit    |
|------------------|---------------------------------------|---|----------------|----------------|---------|
| $I_{LI}$         | Input Leakage Current                 | $V_{CC} = V_{CC \text{ Max.}}$ ,<br>$V_{IN} = V_{CC}$ or $V_{SS}$   |                | + 1.0          | $\mu A$ |
| $I_{LO}$         | Output Leakage Current                | $V_{CC} = V_{CC \text{ Max.}}$ ,<br>$V_{OUT} = V_{CC}$ or $V_{SS}$  |                | + 1.0          | $\mu A$ |
| $I_{CCS}$        | $V_{CC}$ Standby Current              | $V_{CC} = V_{CC \text{ Max.}}$ ,<br>$\overline{CE} = V_{CC} \pm 0.3 \text{ V}$  |                | 25             | $\mu A$ |
| $I_{CC1}$        | $V_{CC}$ Active Read Current          | $V_{CC} = V_{CC \text{ Max.}}$ , $\overline{CE} = V_{IL}$ , $\overline{OE} = V_{IH}$<br>$I_{OUT} = 0 \text{ mA}$ , at 5 MHz |                | 15             | mA      |
| $I_{CC2}$        | $V_{CC}$ Programming Current          | $\overline{CE} = V_{IL}$<br>Programming in Progress   |                | 30             | mA      |
| $I_{PPS}$        | $V_{PP}$ Standby Current              | $V_{PP} = V_{PPL}$  |                | + 1.0          | $\mu A$ |
| $V_{IL}$         | Input Low Voltage                     |   | -0.5           | 0.6            | V       |
| $V_{IH}$         | Input High Voltage                    |   | 2.0            | $V_{CC} + 0.5$ | V       |
| $V_{OL}$         | Output Low Voltage                    | $I_{OL} = 1 \text{ mA}$<br>$V_{CC} = V_{CC \text{ Min.}}$   |                | 0.3            | V       |
| $V_{OH}$         | Output High Voltage                   | $I_{OH} = -100 \mu A$ , $V_{CC} = V_{CC \text{ Min.}}$  | $V_{CC} - 0.3$ |                | V       |
| $V_{ID}$         | $A_9$ Auto Select Voltage             | $A_9 = V_{ID}$  | 11.5           | 13.0           | V       |
| $I_{ID}$         | $A_9$ Auto Select Current             | $A_9 = V_{ID \text{ Max.}}$ ,<br>$V_{CC} = V_{CC \text{ Max.}}$   |                | 35             | $\mu A$ |
| $V_{PPL}$        | $V_{PP}$ during Read-Only Operations  |   | 0.0            | $V_{CC} + 2.0$ | V       |
| $V_{PPH}$        | $V_{PP}$ during Read/Write Operations |   | 12.5           | 13.0           | V       |

## Notes:

1. **Caution:** the Am27LV010 must not be removed from (or inserted into) a socket when  $V_{CC}$  or  $V_{PP}$  is applied.
2.  $I_{CC1}$  is tested with  $\overline{OE} = V_{IH}$  to simulate open outputs.
3. Maximum active power usage is the sum of  $I_{CC}$  and  $I_{PP}$ .

## PIN CAPACITANCE

| Parameter Symbol | Parameter Description             | Test Conditions      | Typ. | Max. | Unit |
|------------------|-----------------------------------|----------------------|------|------|------|
| C <sub>IN</sub>  | Input Capacitance                 | V <sub>IN</sub> = 0  | 8    | 10   | pF   |
| C <sub>OUT</sub> | Output Capacitance                | V <sub>OUT</sub> = 0 | 8    | 12   | pF   |
| C <sub>IN2</sub> | V <sub>PP</sub> Input Capacitance | V <sub>PP</sub> = 0  | 8    | 12   | pF   |

### Notes:

1. Sampled, not 100% tested.
2. Test conditions T<sub>A</sub> = 25°C, f = 1.0 MHz.

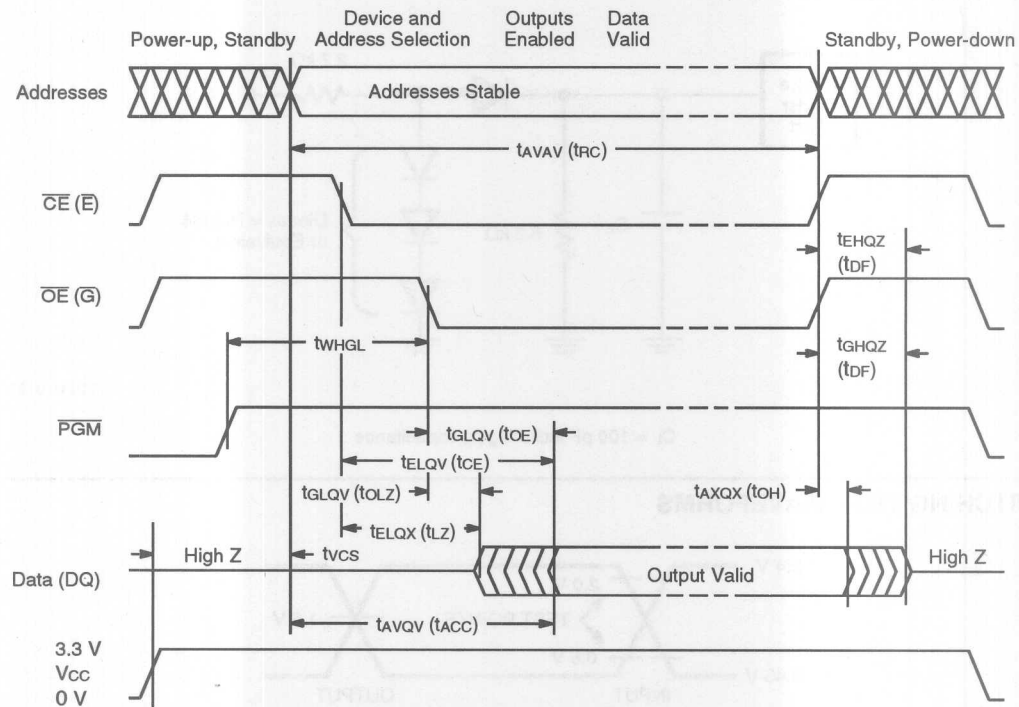
## SWITCHING CHARACTERISTICS over operating range unless otherwise specified

### AC CHARACTERISTICS—Read Only Operation (Note 1)

| Parameter Symbols |                  | Parameter Description                               |              | Am27LV010 |      |      |      |
|-------------------|------------------|---|--------------|-----------|------|------|------|
| JEDEC             | Standard         |   |              | -200      | -250 | -300 | Unit |
| t <sub>AVAV</sub> | t <sub>RC</sub>  | Read Cycle Time                                     | Min.<br>Max. | 200       | 250  | 300  | ns   |
| t <sub>ELQV</sub> | t <sub>CE</sub>  | Chip Enable Access Time                             | Min.<br>Max. | 200       | 250  | 300  | ns   |
| t <sub>AVQV</sub> | t <sub>ACC</sub> | Address Access Time                                 | Min.<br>Max. | 200       | 250  | 300  | ns   |
| t <sub>GLQV</sub> | t <sub>OE</sub>  | Output Enable Access Time                           | Min.<br>Max. | 75        | 100  | 100  | ns   |
| t <sub>ELQX</sub> | t <sub>LZ</sub>  | Chip Enable to Output in Low Z                      | Min.<br>Max. | 0         | 0    | 0    | ns   |
| t <sub>EHQZ</sub> | t <sub>DF</sub>  | Chip Disable to Output in High Z                    | Min.<br>Max. | 35        | 35   | 35   | ns   |
| t <sub>GLQX</sub> | t <sub>OLZ</sub> | Output Enable to Output in Low Z                    | Min.<br>Max. | 0         | 0    | 0    | ns   |
| t <sub>GHQZ</sub> | t <sub>DF</sub>  | Output Disable to Output in High Z                  | Min.<br>Max. | 35        | 35   | 35   | ns   |
| t <sub>AXQX</sub> | t <sub>OH</sub>  | Output Hold from first of Address, CE, or OE Change | Min.<br>Max. | 0         | 0    | 0    | ns   |
| t <sub>VCS</sub>  |                  | V <sub>CC</sub> Set-up Time to Valid Read           | Min.<br>Max. | 50        | 50   | 50   | μs   |

### Notes:

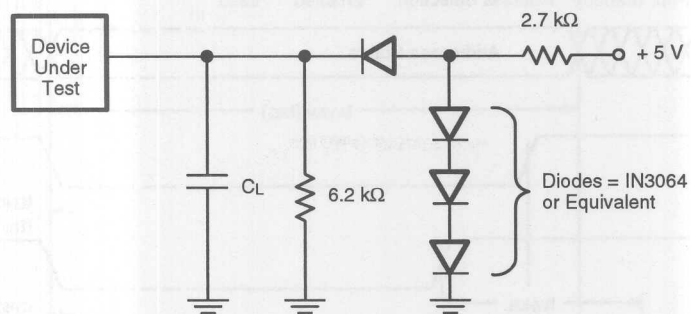
1. Output Load: 1 TTL gate and C<sub>L</sub> = 100 pF, Input Rise and Fall Times: ≤ 10 ns, Input Pulse levels: 0.45 to 2.4 V, Timing Measurement Reference Level - Inputs: 0.6 V and 2 V  
Outputs: 1.5 V
2. t<sub>VCS</sub> is guaranteed by design not tested.



16885A-007A

## AC Waveforms for Read Operations

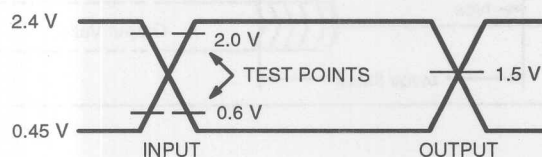
# SWITCHING TEST CIRCUIT



11561-012A

$C_L = 100 \text{ pF}$  including jig capacitance

# SWITCHING TEST WAVEFORMS



All Devices

AC Testing: Inputs are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0". Input pulse rise and fall times are  $\leq 10 \text{ ns}$ .

16357A-002B

**ERASE AND PROGRAMMING PERFORMANCE**

| Parameter             | Limits |               |      | Unit | Comments                       |
|-----------------------|--------|---------------|------|------|--------------------------------|
|                       | Min.   | Typ.          | Max. |      |                                |
| Chip Programming Time |        | 4<br>(Note 1) | 48   | S    | Excludes system-level overhead |

**Note:**

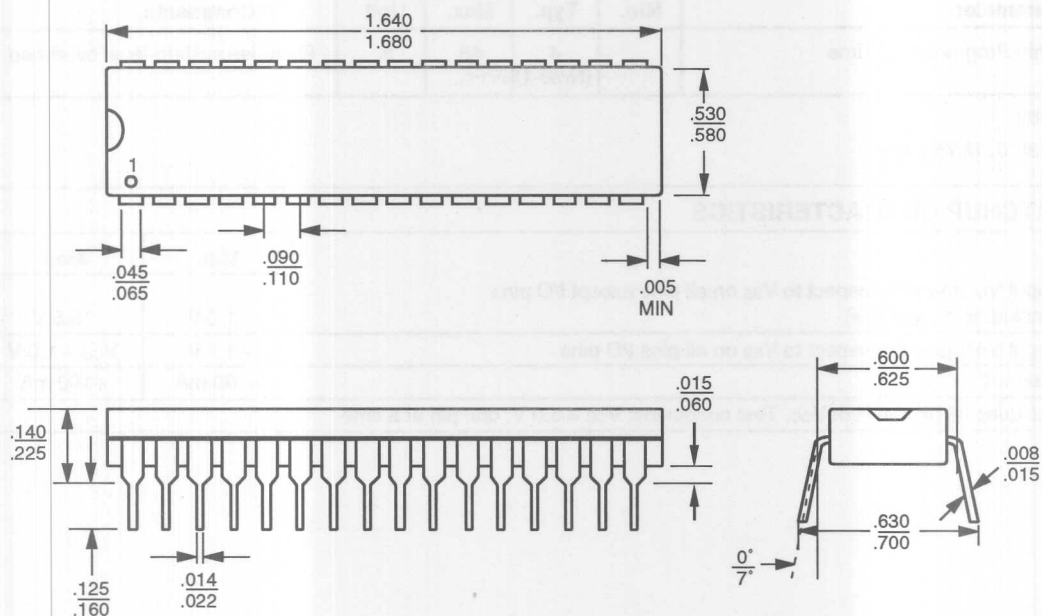
1. 25°C, 12.75 V V<sub>PP</sub>.

**LATCHUP CHARACTERISTICS**

|  | Min.    | Max.                    |
|--|---------|-------------------------|
| Input Voltage with respect to V <sub>SS</sub> on all pins except I/O pins<br>(Including A <sub>9</sub> and V <sub>PP</sub> ) | -1.0 V  | 13.5 V                  |
| Input Voltage with respect to V <sub>SS</sub> on all pins I/O pins   | -1.0 V  | V <sub>CC</sub> + 1.0 V |
| Current  | -100 mA | +100 mA                 |
| Includes all pins except V <sub>CC</sub> . Test conditions: V <sub>CC</sub> = 5.0 V, one pin at a time.                      |         |                         |

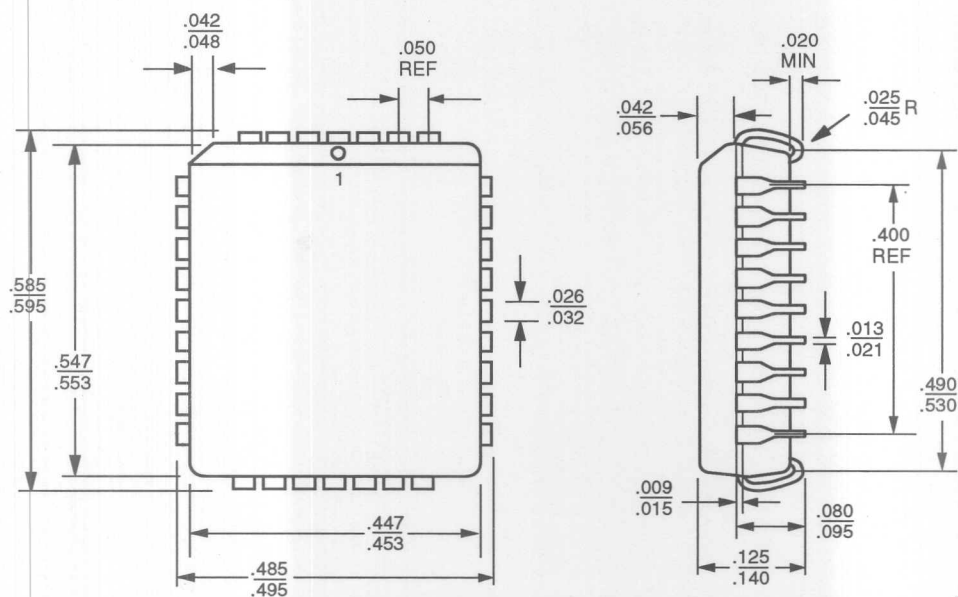
PHYSICAL DIMENSIONS\*

PD 032



12416B

PL 032



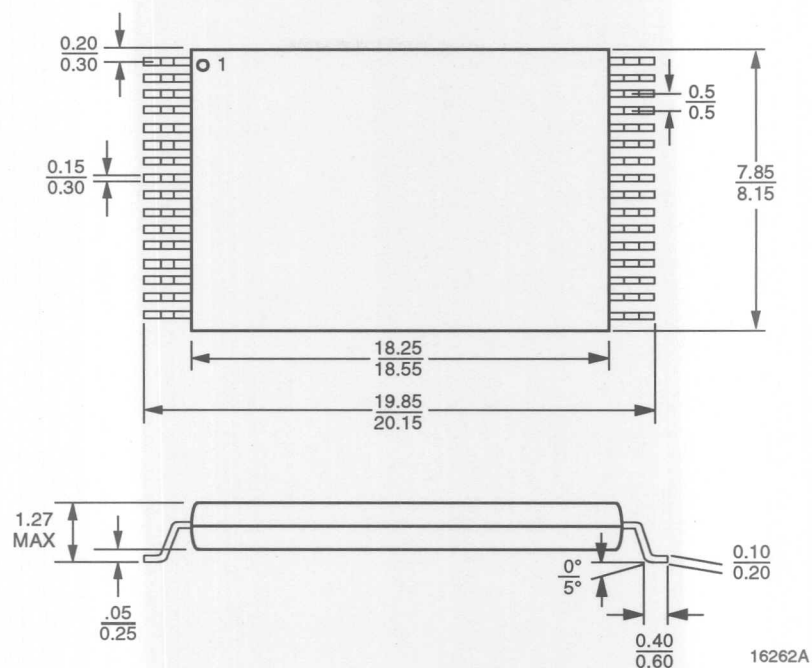
06971C

\*For reference only. All dimensions are measured in inches, unless otherwise noted. BSC is an ANSI standard for Basic Space Centering.

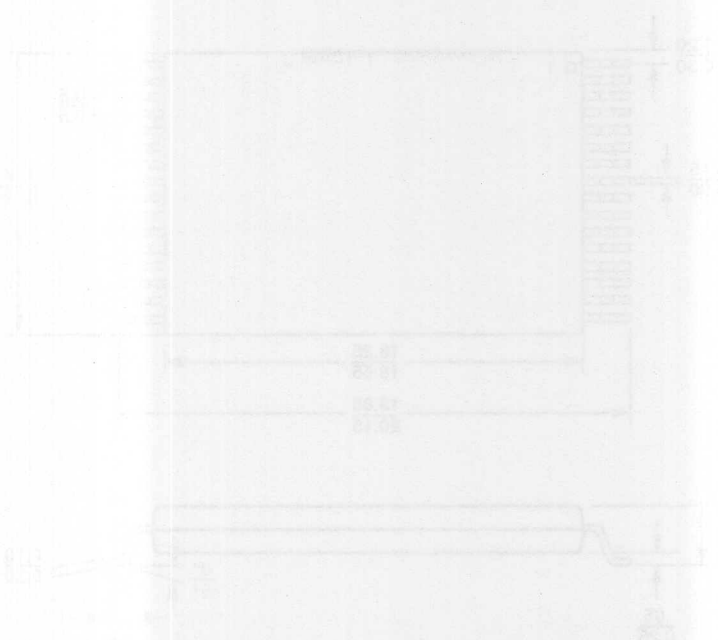


# PHYSICAL DIMENSIONS\*

## TS 032



\*For reference only. All dimensions are measured in inches, unless otherwise noted. BSC is an ANSI standard for Basic Space Centering.





# Am27LV020

262,144 x 8-Bit CMOS Low Voltage, One Time Programmable Memory

## DISTINCTIVE CHARACTERISTICS

- **3.3 V  $\pm$  0.3 V  $V_{CC}$  read operation**
- **High performance at 3.3 V $_{CC}$** 
  - 200 ns maximum access time
- **Low power consumption**
  - 90  $\mu$ W maximum standby power
  - 25  $\mu$ A maximum standby current
  - 54 mW maximum power at 5 MHz
  - 15 mA maximum current at 5 MHz
  - No data retention power
- **Industry standard packaging**
  - 32-pin PLCC
  - 32-pin Thin Small Outline Package
  - 32-pin Plastic DIP
- **Program voltage  $12.75 \pm .25$  V**
- **Latch-up protected to 100 mA from  $-1$  V to  $V_{CC} + 1$  V**
- **Flashrite™ programming**
  - 10  $\mu$ s typical byte-program
  - Less than 3 seconds typical chip program
- **Advanced CMOS memory technology**
  - Low cost single transistor memory cell

## GENERAL DESCRIPTION

The Am27LV020 device is a low voltage, low power, CMOS 256K x 8 One Time Programmable (OTP) non-volatile memory.

Maximum power consumption in standby mode is 90  $\mu$ W. If the device is constantly accessed at 5 MHz, then maximum power consumption increases to 54 mW. These power ratings are significantly lower than typical EPROM devices. Since power consumption is proportional to voltage squared, 3.3 V devices typically consume at least 57% less power than 5.0 V devices.

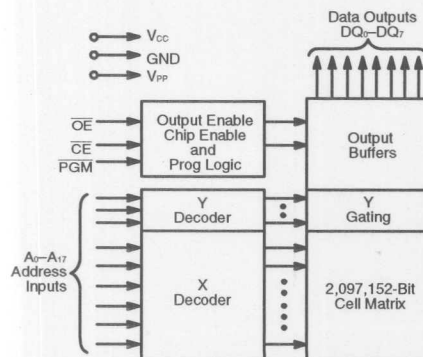
The Am27LV020 typically draws 10 mA of current enabling 200 ns read operations. Typical power consumption under these conditions equals 33 mW. This "high

performance", low voltage device is ideal for BIOS storage in portable computing applications and control code storage in portable digital cellular phone applications. Low voltage CMOS designs require less operating power and hence dramatically increases the usable operating life of battery powered systems.

The Am27LV020 is packaged in a 32-pin PLCC, Plastic DIP and Thin Small Outline Package (TSOP) versions. It is designed to be programmed in standard EPROM programmers.

The highest degree of latch-up protection is achieved with AMD's proprietary non-epi process. Latch-up protection is provided for stresses up to 100 milliamps on address and data pins from  $-1$  V to  $V_{CC} + 1$  V.

## BLOCK DIAGRAM



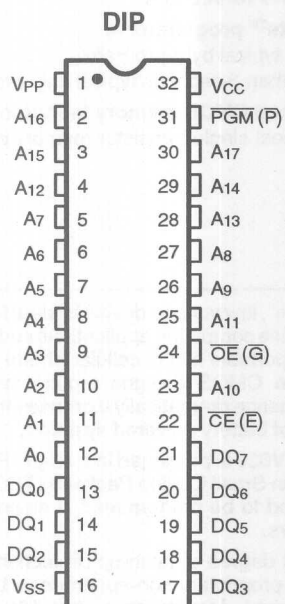
11507-001B

This document contains information on a product under development at Advanced Micro Devices, Inc. The information is intended to help you evaluate this product. AMD reserves the right to change or discontinue work on this proposed product without notice.

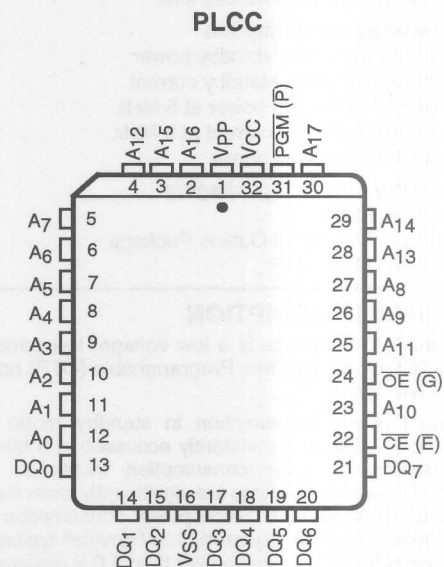
PRODUCT SELECTOR GUIDE

| Family Part No.                       | Am27LV020 |      |      |
|---------------------------------------|-----------|------|------|
| Ordering Part No:                     |           |      |      |
| $\pm 0.3 \text{ V } V_{CC}$ Tolerance | -200      | -250 | -300 |
| Max Access Time (ns)                  | 200       | 250  | 300  |
| $\overline{CE}$ (E) Access (ns)       | 200       | 250  | 300  |
| $\overline{OE}$ (G) Access (ns)       | 75        | 100  | 100  |

CONNECTION DIAGRAMS



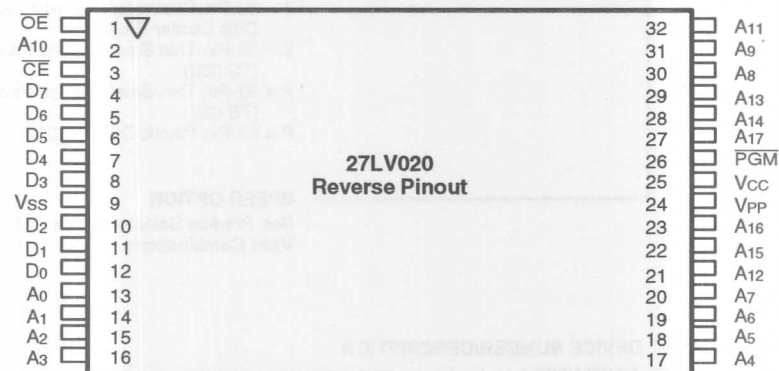
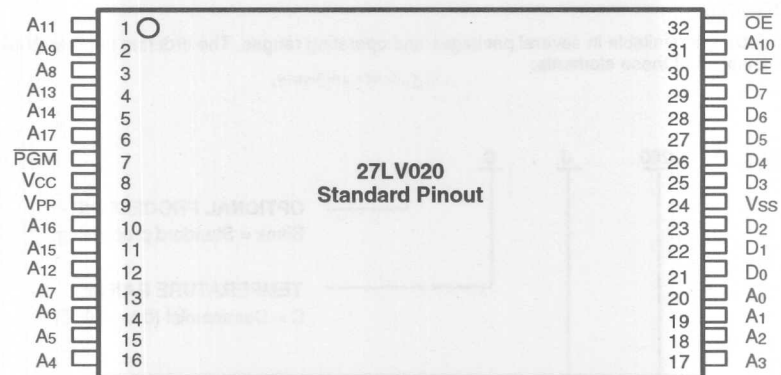
14727-002B



14727-003B

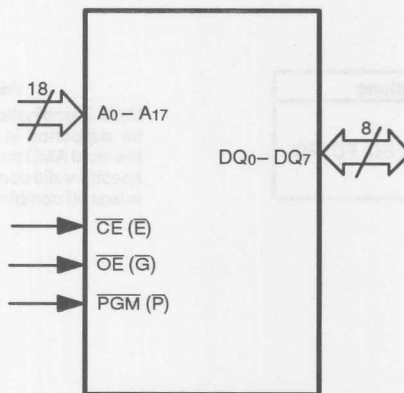
**Note:** Pin 1 is marked for orientation.

## TSOP PACKAGES



27LV020 256K x 8 OTP in 32 Lead TSOP

## LOGIC SYMBOL

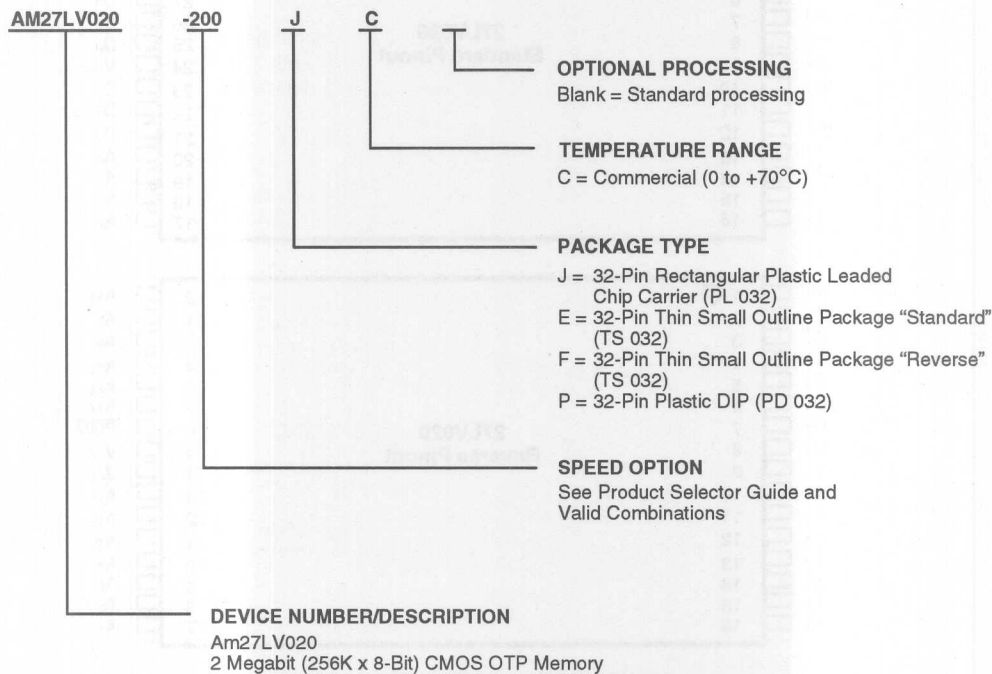


11559-004B

## ORDERING INFORMATION

### Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of these elements:



| Valid Combinations |                |
|--------------------|----------------|
| Am27LV020-200      | JC, EC, FC, PC |
| Am27LV020-250      |                |
| Am27LV020-300      |                |

#### Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

**PIN DESCRIPTION****V<sub>PP</sub>**

Power supply for programming.

**V<sub>CC</sub>**

Power supply for device operation. (Read:  $V_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ , Program:  $V_{CC} = 5.0\text{ V} \pm 10\%$ )

**V<sub>SS</sub>**

Ground

**NC**

No Connect-corresponding pin is not connected internally to the die.

**A<sub>0</sub> – A<sub>17</sub>**

Address Inputs for memory locations.

**DQ<sub>0</sub> – DQ<sub>7</sub>**

Data Inputs during memory program cycles. Internal latches hold data during program cycles. Data Outputs during memory read cycles.

 **$\overline{\text{CE}}$  ( $\overline{\text{E}}$ )**

The Chip Enable active low input activates the chip's control logic and input buffers. Chip Enable high will deselect the device and operates the chip in stand-by mode.

 **$\overline{\text{OE}}$  ( $\overline{\text{G}}$ )**

The Output Enable active low input gates the outputs of the device through the data buffers during memory read cycles.

 **$\overline{\text{PGM}}$  ( $\overline{\text{P}}$ )**

The Program Enable active low input controls the program function of the memory array.

**BASIC PRINCIPLES**

The Am27LV020 supports programming operations using a fixed  $12.75 \pm 0.25\text{ V}$  power supply.

**Read Only Memory**

Without high  $V_{PP}$  voltage, the Am27LV020 functions as a read only memory and operates like a standard EPROM. The control inputs still manage traditional read, standby, output disable, and Auto select modes.

**Programming**

These devices are programmable on standard PROM programmer equipment.

Please contact Advanced Micro Devices for PROM programmer information.

**FUNCTIONAL DESCRIPTION****Description Of User Modes****Table 1. Am27LV020 User Bus Operations**

| Operation |                               | $\overline{\text{CE}}$<br>( $\overline{\text{E}}$ ) | $\overline{\text{OE}}$<br>( $\overline{\text{G}}$ ) | $\overline{\text{WE}}$<br>( $\overline{\text{W}}$ ) | V <sub>PP</sub><br>(Note 1) | A <sub>0</sub>  | A <sub>9</sub>              | I/O              |
|-----------|-------------------------------|---|---|---|-----------------------------|-----------------|-----------------------------|------------------|
| Read-Only | Read                          | V <sub>IL</sub>                                     | V <sub>IL</sub>                                     | X   | V <sub>PPL</sub>            | A <sub>0</sub>  | A <sub>9</sub>              | D <sub>OUT</sub> |
|           | Standby                       | V <sub>IH</sub>                                     | X   | X   | V <sub>PPL</sub>            | X               | X                           | HIGH Z           |
|           | Output Disable                | V <sub>IL</sub>                                     | V <sub>IH</sub>                                     | V <sub>IH</sub>                                     | V <sub>PPL</sub>            | X               | X                           | HIGH Z           |
|           | Auto-select Manufacturer Code | V <sub>IL</sub>                                     | V <sub>IL</sub>                                     | V <sub>IH</sub>                                     | V <sub>PPL</sub>            | V <sub>IL</sub> | V <sub>ID</sub><br>(Note 2) | CODE<br>(01H)    |
|           | Auto-select Device Code       | V <sub>IL</sub>                                     | V <sub>IL</sub>                                     | V <sub>IH</sub>                                     | V <sub>PPL</sub>            | V <sub>IH</sub> | V <sub>ID</sub><br>(Note 2) | CODE<br>(2CH)    |

**Legend:**

X = Don't care, where Don't Care is either V<sub>IL</sub> or V<sub>IH</sub> levels, V<sub>PPL</sub> =  $V_{PP} < V_{CC} + 2\text{ V}$ , See DC Characteristics for voltage levels of V<sub>PPH</sub>,  $0\text{ V} < A_n < V_{CC} + 2\text{ V}$ , (normal CMOS input levels, where n = 0 or 9).

**Notes:**

1. V<sub>PPL</sub> may be grounded, connected with a resistor to ground, or  $\leq V_{CC} + 2.0\text{ V}$ . V<sub>PPH</sub> is the programming voltage specified for the device. Refer to the DC characteristics. When V<sub>PP</sub> = V<sub>PPL</sub>, memory contents can be read but not written.
2.  $11.5 \leq V_{ID} \leq 13.0\text{ V}$ ,  $V_{CC} = 5.0\text{ V} \pm 10\%$

## READ ONLY MODE

$$V_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}$$

### Read

The Am27LV020 functions as a read only memory. The Am27LV020 has two control functions. Both must be satisfied in order to output data.  $\overline{CE}$  controls power to the device. This pin should be used for specific device selection.  $\overline{OE}$  controls the device outputs and should be used to gate data to the output pins if a device is selected.

Address access time  $t_{ACC}$  is equal to the delay from stable addresses to valid output data. The chip enable access time  $t_{CE}$  is the delay from stable addresses and stable  $\overline{CE}$  to valid data at the output pins. The output enable access time is the delay from the falling edge of  $\overline{OE}$  to valid data at the output pins (assuming the addresses have been stable at least  $t_{ACC} - t_{OE}$ ).

### Standby Mode

The Am27LV020 has one standby mode. The CMOS standby mode ( $\overline{CE}$  input held at  $V_{CC} \pm 0.5\text{V}$ ), consumes less than 25  $\mu\text{A}$  of current. When in the standby mode the outputs are in a high impedance state, independent of the  $\overline{OE}$  input.

If the device is deselected during programming, or program verification, the device will draw active current until the operation is terminated.

### Output Disable

Output from the device is disabled when  $\overline{OE}$  is at a logic high level. When disabled, output pins are in a high impedance state.

### Auto Select

The Am27LV020 can be programmed in a standard PROM programmer.

The Auto select mode allows the reading out of a binary code from the device that will identify its manufacturer and type. This mode is intended for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm.

### Programming In A PROM Programmer

To activate this mode, the programming equipment must force  $V_{ID}$  (11.5V to 13.0V) on address  $A_9$ . Two identifier bytes may then be sequenced from the device outputs by toggling address  $A_0$  from  $V_{IL}$  to  $V_{IH}$ . All other address lines must be held at  $V_{IL}$ , and  $V_{PP}$  must be less than or equal to  $V_{CC} + 2.0\text{V}$  while using this Auto select mode. Byte 0 ( $A_0 = V_{IL}$ ) represents the manufacturer code and byte 1 ( $A_0 = V_{IH}$ ) the device identifier code. For the Am27LV020 these two bytes are given in the table below. All identifiers for manufacturer and device codes will exhibit odd parity with the MSB ( $DQ_7$ ) defined as the parity bit.

Table 2. Am27LV020 Auto Select Code

| Type              | $A_0$    | Code (HEX) | $DQ_7$ | $DQ_6$ | $DQ_5$ | $DQ_4$ | $DQ_3$ | $DQ_2$ | $DQ_1$ | $DQ_0$ |
|-------------------|----------|------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Manufacturer Code | $V_{IL}$ | 01         | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 1      |
| Device Code       | $V_{IH}$ | 2C         | 0      | 0      | 1      | 0      | 1      | 1      | 0      | 0      |



**ABSOLUTE MAXIMUM RATINGS**

|  |                   |
|--|-------------------|
| Storage Temperature                                |                   |
| Plastic Packages                                   | −65°C to +125°C   |
| Ambient Temperature                                |                   |
| with Power Applied                                 | −55°C to +125°C   |
| Voltage with Respect To Ground. All pins           |                   |
| except A <sub>9</sub> and V <sub>PP</sub> (Note 1) | −2.0 V to +7.0 V  |
| V <sub>CC</sub> (Note 1)                           | −2.0 V to +7.0 V  |
| A <sub>9</sub> (Note 2)                            | −2.0 V to +14.0 V |
| V <sub>PP</sub> (Note 2)                           | −2.0 V to +14.0 V |
| Output Short Circuit Current (Note 3)              | 200 mA            |

**Notes:**

1. Minimum DC voltage on input or I/O pins is −0.5 V. During voltage transitions, inputs may overshoot V<sub>SS</sub> to −2.0 V for periods of up to 20 ns. Maximum DC voltage on output and I/O pins is V<sub>CC</sub> + 0.5 V. During voltage transitions, outputs may overshoot to V<sub>CC</sub> + 2.0 V for periods up to 20 ns.
2. During programming operations only. Minimum DC input voltage on A<sub>9</sub> and V<sub>PP</sub> pins is −0.5V. During voltage transitions, A<sub>9</sub> and V<sub>PP</sub> may overshoot V<sub>SS</sub> to −2.0 V for periods of up to 20 ns. Maximum DC input voltage on A<sub>9</sub> and V<sub>PP</sub> is +13.5 V which may overshoot to 14.0 V for periods up to 20 ns.
3. No more than one output shorted at a time. Duration of the short circuit should not be greater than one second.

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure of the device to absolute maximum rating conditions for extended periods may affect device reliability.

**OPERATING RANGES****Commercial (C) Devices**

Case Temperature (T<sub>C</sub>) . . . . . 0°C to +70°C

**V<sub>CC</sub> Supply Voltages**

V<sub>CC</sub> for Am27LV020 . . . . . +3.0 V to +3.6 V

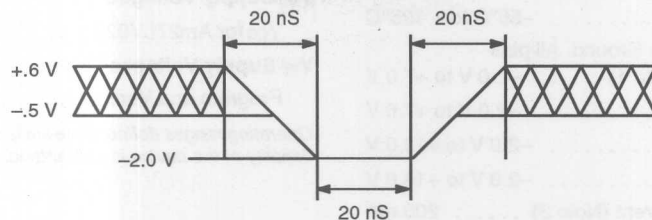
**V<sub>PP</sub> Supply Voltages**

Program and Verify . . . . . +12.5 V to +13 V

*Operating ranges define those limits between which the functionality of the device is guaranteed.*

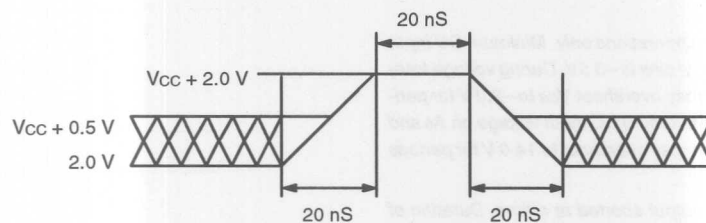
# MAXIMUM OVERSHOOT

## Maximum Negative Input Overshoot



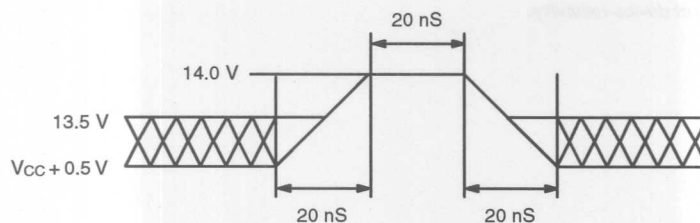
11561-009B

## Maximum Positive Input Overshoot



11561-010A

## Maximum $V_{PP}$ Overshoot



11561-011A

## DC CHARACTERISTICS-CMOS COMPATIBLE

| Parameter Symbol | Parameter Description                 | Test Conditions   | Min.           | Max.           | Unit    |
|------------------|---------------------------------------|---|----------------|----------------|---------|
| $I_{LI}$         | Input Leakage Current                 | $V_{CC} = V_{CC \text{ Max.}}$ ,<br>$V_{IN} = V_{CC}$ or $V_{SS}$   |                | + 1.0          | $\mu A$ |
| $I_{LO}$         | Output Leakage Current                | $V_{CC} = V_{CC \text{ Max.}}$ ,<br>$V_{OUT} = V_{CC}$ or $V_{SS}$  |                | + 1.0          | $\mu A$ |
| $I_{CCS}$        | $V_{CC}$ Standby Current              | $V_{CC} = V_{CC \text{ Max.}}$<br>$\overline{CE} = V_{CC} \pm 0.3 \text{ V}$  |                | 25             | $\mu A$ |
| $I_{CC1}$        | $V_{CC}$ Active Read Current          | $V_{CC} = V_{CC \text{ Max.}}$ , $\overline{CE} = V_{IL}$ , $\overline{OE} = V_{IH}$<br>$I_{OUT} = 0 \text{ mA}$ , at 5 MHz |                | 15             | mA      |
| $I_{CC2}$        | $V_{CC}$ Programming Current          | $\overline{CE} = V_{IL}$<br>Programming in Progress   |                | 30             | mA      |
| $I_{PPS}$        | $V_{PP}$ Standby Current              | $V_{PP} = V_{PPL}$  |                | + 1.0          | $\mu A$ |
| $V_{IL}$         | Input Low Voltage                     |   | -0.5           | 0.6            | V       |
| $V_{IH}$         | Input High Voltage                    |   | 2.0            | $V_{CC} + 0.5$ | V       |
| $V_{OL}$         | Output Low Voltage                    | $I_{OL} = 1 \text{ mA}$<br>$V_{CC} = V_{CC \text{ Min.}}$   |                | 0.3            | V       |
| $V_{OH}$         | Output High Voltage                   | $I_{OH} = -100 \mu A$ , $V_{CC} = V_{CC \text{ Min.}}$  | $V_{CC} - 0.3$ |                | V       |
| $V_{ID}$         | $A_9$ Auto Select Voltage             | $A_9 = V_{ID}$  | 11.5           | 13.0           | V       |
| $I_{ID}$         | $A_9$ Auto Select Current             | $A_9 = V_{ID \text{ Max.}}$<br>$V_{CC} = V_{CC \text{ Max.}}$   |                | 35             | $\mu A$ |
| $V_{PPL}$        | $V_{PP}$ during Read-Only Operations  |   | 0.0            | $V_{CC} + 2.0$ | V       |
| $V_{PPH}$        | $V_{PP}$ during Read/Write Operations |   | 12.5           | 13.0           | V       |

## Notes:

1. **Caution:** the Am27LV020 must not be removed from (or inserted into) a socket when  $V_{CC}$  or  $V_{PP}$  is applied.
2.  $I_{CC1}$  is tested with  $\overline{OE} = V_{IH}$  to simulate open outputs.
3. Maximum active power usage is the sum of  $I_{CC}$  and  $I_{PP}$ .

## PIN CAPACITANCE

| Parameter Symbol | Parameter Description             | Test Conditions      | Typ. | Max. | Unit |
|------------------|-----------------------------------|----------------------|------|------|------|
| C <sub>IN</sub>  | Input Capacitance                 | V <sub>IN</sub> = 0  | 8    | 10   | pF   |
| C <sub>OUT</sub> | Output Capacitance                | V <sub>OUT</sub> = 0 | 8    | 12   | pF   |
| C <sub>IN2</sub> | V <sub>PP</sub> Input Capacitance | V <sub>PP</sub> = 0  | 8    | 12   | pF   |

### Notes:

1. Sampled, not 100% tested.
2. Test conditions T<sub>A</sub> = 25°C, f = 1.0 MHz.

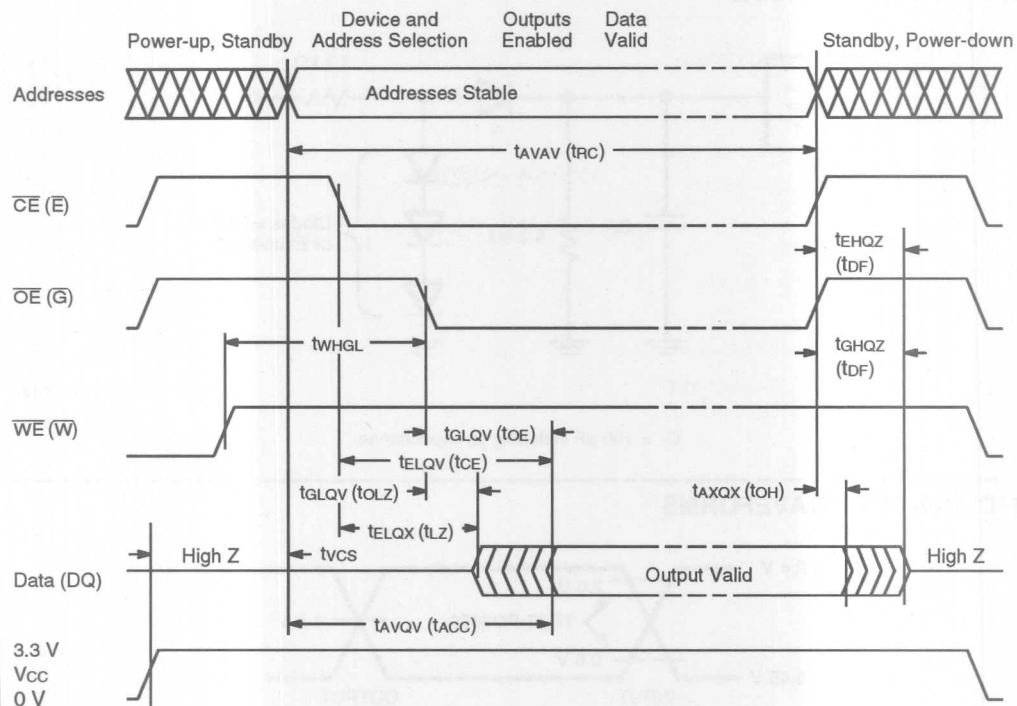
## SWITCHING CHARACTERISTICS over operating range unless otherwise specified

### AC CHARACTERISTICS—Read Only Operation (Note 1)

| Parameter Symbols |                  | Parameter Description                               |              | Am27LV020 |      |      |      |
|-------------------|------------------|---|--------------|-----------|------|------|------|
| JEDEC             | Standard         |   |              | -200      | -250 | -300 | Unit |
| t <sub>AVAV</sub> | t <sub>RC</sub>  | Read Cycle Time                                     | Min.<br>Max. | 200       | 250  | 300  | ns   |
| t <sub>ELQV</sub> | t <sub>CE</sub>  | Chip Enable Access Time                             | Min.<br>Max. | 200       | 250  | 300  | ns   |
| t <sub>AVQV</sub> | t <sub>ACC</sub> | Address Access Time                                 | Min.<br>Max. | 200       | 250  | 300  | ns   |
| t <sub>GLQV</sub> | t <sub>OE</sub>  | Output Enable Access Time                           | Min.<br>Max. | 75        | 100  | 100  | ns   |
| t <sub>ELQX</sub> | t <sub>LZ</sub>  | Chip Enable to Output in Low Z                      | Min.<br>Max. | 0         | 0    | 0    | ns   |
| t <sub>EHQZ</sub> | t <sub>DF</sub>  | Chip Disable to Output in High Z                    | Min.<br>Max. | 35        | 35   | 35   | ns   |
| t <sub>GLQX</sub> | t <sub>OLZ</sub> | Output Enable to Output in Low Z                    | Min.<br>Max. | 0         | 0    | 0    | ns   |
| t <sub>GHQZ</sub> | t <sub>DF</sub>  | Output Disable to Output in High Z                  | Min.<br>Max. | 35        | 35   | 35   | ns   |
| t <sub>AXQX</sub> | t <sub>OH</sub>  | Output Hold from first of Address, CE, or OE Change | Min.<br>Max. | 0         | 0    | 0    | ns   |
| t <sub>VCS</sub>  |                  | V <sub>CC</sub> Set-up Time to Valid Read           | Min.<br>Max. | 50        | 50   | 50   | μs   |

### Notes:

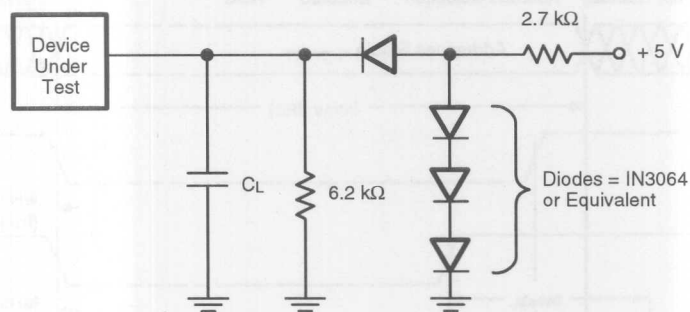
1. Output Load: 1 TTL gate and C<sub>L</sub> = 100 pF, Input Rise and Fall Times: ≤ 10 ns, Input Pulse levels: 0.45 to 2.4 V, Timing Measurement Reference Level - Inputs: 0.6 V and 2 V  
Outputs: 1.5 V
2. t<sub>VCS</sub> is guaranteed by design not tested.



11561-013C

AC Waveforms for Read Operations

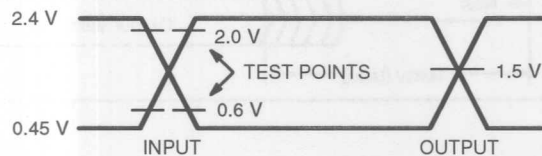
# SWITCHING TEST CIRCUIT



11561-012A

$C_L = 100 \text{ pF}$  including jig capacitance

# SWITCHING TEST WAVEFORMS



All Devices

AC Testing: Inputs are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0". Input pulse rise and fall times are  $\leq 10 \text{ ns}$ .

16357A-002B

**ERASE AND PROGRAMMING PERFORMANCE**

| Parameter             | Limits |               |      | Unit | Comments                       |
|-----------------------|--------|---------------|------|------|--------------------------------|
|                       | Min.   | Typ.          | Max. |      |                                |
| Chip Programming Time |        | 4<br>(Note 1) | 48   | S    | Excludes system-level overhead |

**Note:**

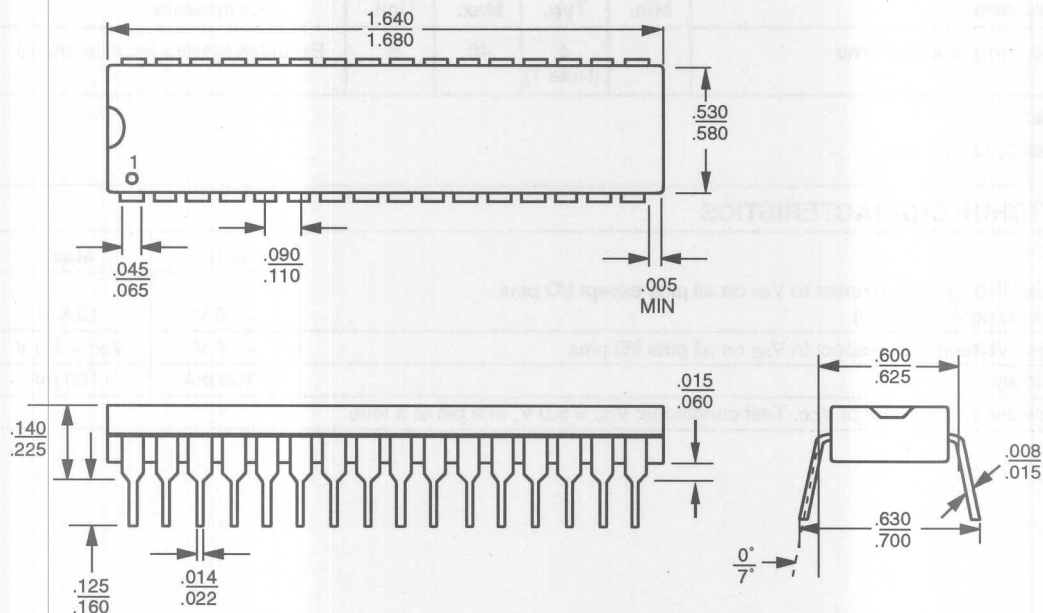
1. 25°C, 12.75 V  $V_{PP}$ .

**LATCHUP CHARACTERISTICS**

|   | Min.    | Max.             |
|---|---------|------------------|
| Input Voltage with respect to $V_{SS}$ on all pins except I/O pins<br>(Including $A_9$ and $V_{PP}$ ) | -1.0 V  | 13.5 V           |
| Input Voltage with respect to $V_{SS}$ on all pins I/O pins   | -1.0 V  | $V_{CC} + 1.0$ V |
| Current   | -100 mA | +100 mA          |
| Includes all pins except $V_{CC}$ . Test conditions: $V_{CC} = 5.0$ V, one pin at a time.             |         |                  |

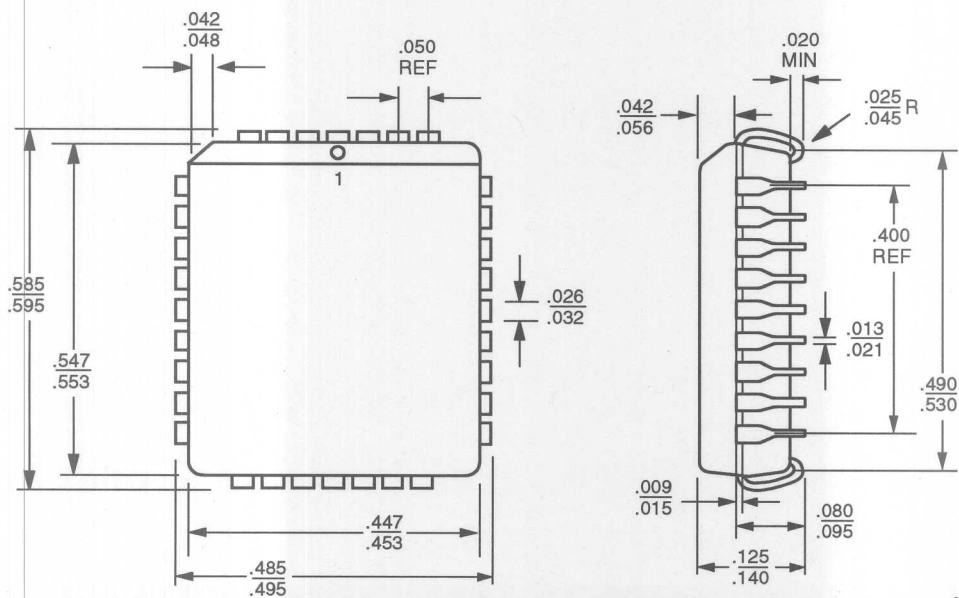
PHYSICAL DIMENSIONS\*

PD 032



12416B

PL 032



06971C

\*For reference only. All dimensions are measured in inches, unless otherwise noted. BSC is an ANSI standard for Basic Space Centering.



## PHYSICAL DIMENSIONS

TS 032

